# ABSTRACT


Title of dissertation: CIRCUIT DESIGN AND ROUTING FOR FIELD PROGRAMMABLE ANALOG ARRAYS

Ji Luo, Doctor of Philosophy, 2005

Dissertation directed by: Professor Joseph Bernstein
Professor Martin Peckerar
Department of Electrical & Computer Engineering



Accurate, low-cost, rapid-prototyping techniques for analog circuits have been a long awaited dream for analog designers. However, due to the inherent nature of analog system, design automation in analog domain is very difficult to realize, and field programmable analog arrays (FPAA) have not achieved the same success as FPGAs in the digital domain. This results from several factors, including the lack of supporting CAD tools, small circuit density, low speed and significant parasitic effect from the fixed routing wires. These factors are all related to each other, making the design of a high performance FPAA a multi-dimension problem. Among others, a critical reason behind these difficulties is the non-ideal programming technology, which contributes a large portion of parasitics into the sensitive analog system, thus degrades the system performance.

This work is trying to attack these difficulties with development of a laser field programmable analog array (LFPAA). There are two parts of work involved,

routing for FPAA and analog IC building block design. To facilitate the router development and provide a platform for FPAA application development, a generic arrayed based FPAA architecture and a flexible CAB topology were proposed. The routing algorithm was based on a modified and improved pathfinder negotiated routing algorithm, and was implemented in C for a prototype FPAA. The parasitic constraints for performance analog routing were also investigated and solutions were proposed. In the area of analog circuit design, a novel differential difference op amp was invented as the core building block. Two bandgap circuits including a low voltage version were developed to generate a stable reference voltage for the FPAA. Based on the proposed FPAA architecture, several application examples were demonstrated. The results show the flexible functionality of the FPAA. Moreover, various laser Makelink test structures were studied on different CMOS processes and BiCMOS copper process. Laser Makelink proves to be a powerful programming technology for analog IC design. A novel laser Makelink trimming method was invented to reduce the op amp offset. The application of using laser Makelink to reconfigure the analog circuit blocks was presented.

# CIRCUIT DESIGN AND ROUTING FOR FIELD PROGRAMMABLE ANALOG ARRAYS

by

Ji Luo

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2005

Advisory Committee:

Professor Joseph Bernstein, Advisor
Professor Martin Peckerar, Advisor
Professor Neil Goldsman
Professor Pamela Abshire
Professor Ali Mosleh

# DEDICATION

To my parents

# ACKNOWLEDGMENTS

One would be lucky enough to have an exceptional advisor. I have had two. Six years ago, Professor Bernstein put me onto the right track toward this Ph.D. degree. He shared with me his expertise in the academic area as well as his wisdom about life. He has a unique angle of view when facing the tough problems and can quickly grasp the key point. I hope I did learn a little from him. Two and a half years ago, Professor Peckerar founded the Analog Systems Design Laboratory (ASDL). It is a privilege for me to join ASDL since the very beginning and to further conduct my doctorate work under his guidance. He is sharp, energetic and extremely knowledgeable in almost every area. I not only improved my circuit design techniques, but also learned a lot from him in the areas of device physics and semiconductor processing. He has always made himself available for help or advice. So, first of all, I want to thank them for their kindness, enthusiasm, and support. I am very proud of being their student.

I would like also to thank other professors who served in my dissertation committee. In some sense, Professor Neil Goldsman is my "un-official" advisor, and one of my favorite instructors in the ECE department. I'm very grateful for his help through the years. Professor Abshire has always been gracious and helpful since I knew her. I benefit from some of the classic papers she collected, and thank her for the comments on my Ph.D. proposal and the help from her research group. I got

to know Professor Mosleh six years ago when he taught me a mathematics class. As Director of the Reliability Engineering Program, he has a very tight schedule. I sincerely appreciate him for spending his valuable time reading my dissertation and serving in the committee.

The path to completing this dissertation has included the discovery of new friends and colleagues. I want to thank all the fellows in the ASDL and Microelectronics Reliability group. They have made my graduate school experience a cherished one. A special "thank you" goes to Dr. J. Ari Tuchman for managing various projects. It's a precious experience to work with him.

Furthermore, I would like to acknowledge the University of Maryland Graduate School for providing me two-year Fellowship. Thank Professor Bernstein and Professor Peckerar for providing me the research assistantship through various funding sources.

Finally, I am deeply indebted to my family for supporting me every step along the journey. I thank my sister, Chun, for taking care of our parents and for encouraging me to pursue this doctorate degree. Thank my son Kevin for inspiring me and bringing me a lot of happiness. I'm extremely grateful to my dear wife, Jing. We have been walking through some of the hardest time together. Her love, encouragement and care make this dissertation as much hers as it is mine. My parents have breathlessly awaited this dissertation, and they deserve every single credit of any of my achievements. Thank you, Mom and Dad!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

# Chapter 1

# Introduction to FPAA

## 1.1 Why Analog?

Since the 1980s digital signal processing algorithms have become increasingly powerful. With today's advanced CMOS VLSI technology (both TI and STmicroelectronics have successfully commercialized the 65 $nm$ CMOS process [1], [2]), millions of transistors can be integrated into a tiny silicon chip. These high density and powerful digital ICs have made many functions that traditionally were realized in analog form are now easily implemented in the digital domain. This seems to announce the demise of analog circuit. But, why are analog designs still in such great demand?

After all, the real world is analog. Physical properties such as sound, light, temperature, position, speed, pressure, etc., are all "analog signals". Analog circuits play an extremely important role in bringing the "analog world" together with the "digital world". The real world signals need to be conditioned before being processed, either in the digital or analog domain, before driving an analog output. Therefore analog circuit blocks find broad applications including signal processing and conditioning, power management ICs, industry control, function generation,

A/D, D/A converters ... In fact, analog circuits as the interfacing blocks exist in almost every digital IC. "For every dollar spent on microprocessors, another $1.50 is required to create an interface to the rest of the system" [3]. Databeans, Inc. estimates that the analog semiconductor market was worth about $31 billion in 2004. Following a relatively flat year in 2005, this market is expected to rebound significantly in 2006, with up to 17 percent growth [4].

## 1.2 What is a Field Programmable Analog Array?

An important advantage of digital ICs has been their relative ease of design. Figure 1 is a typical digital system design flow [5]. Many CAD compatible digital IC design methodologies have been developed. For example, a standard ASIC design flow includes hardware behavioral description (VHDL/Verilog), design synthesis and optimization, place and route and final fabrication. When time-to-market and cost are primary concerns, the above flow can be implemented through Field Programmable Gate Arrays (FPGAs). On the contrary, analog design still features an intuitive and manual approach. Its design automation is very difficult to realize. The first-pass (analog) silicon depends heavily on the designer's experience, and the design cycle for a successful analog IC is very long.

It's well known that many (digital) ASICs can be quickly implemented and verified by FPGAs with appropriate programming. Due to their low non-recurring engineering (NRE) costs, short time-to-market, ease of design and low testing costs,

FPGAs have become the most popular ASIC solution [6]. Naturally, we may ask: can we have a field programmable analog array (FPAA) as its digital counter part?

In general, an FPAA is a monolithic collection of analog building blocks (configurable analog blocks, i.e., CABs), a programmable routing network used for passing signals between CABs, and a block of memory (for SRAM based FPAA) storing configuration data which is used to define both the functions and structures. Alternatively, the circuit topologies and routing structures may be defined by other methods such as antifuse programming technologies. A commercial FPAA chip (AN10E40) layout is shown in Figure 2. It contains a 4 x 5 CAB array, an interconnect network and 13 I/O blocks. A configuration bit stream stored in the on-chip SRAM is used to configure the topology [7]. Each CAB can implement a number of analog signal processing functions such as amplification, integration, differentiation, addition, subtraction, multiplication, comparison, log, and exponential. The interconnection network routes signals from one CAB to another, and to and from the I/O blocks.

Laser Field Programmable Analog Array, i.e., LFPAA, is a variant of FPAA. All the switches of an LFPAA are implemented with Laser Makelink technology. LFPAAs are programmed with an infrared (IR) laser.

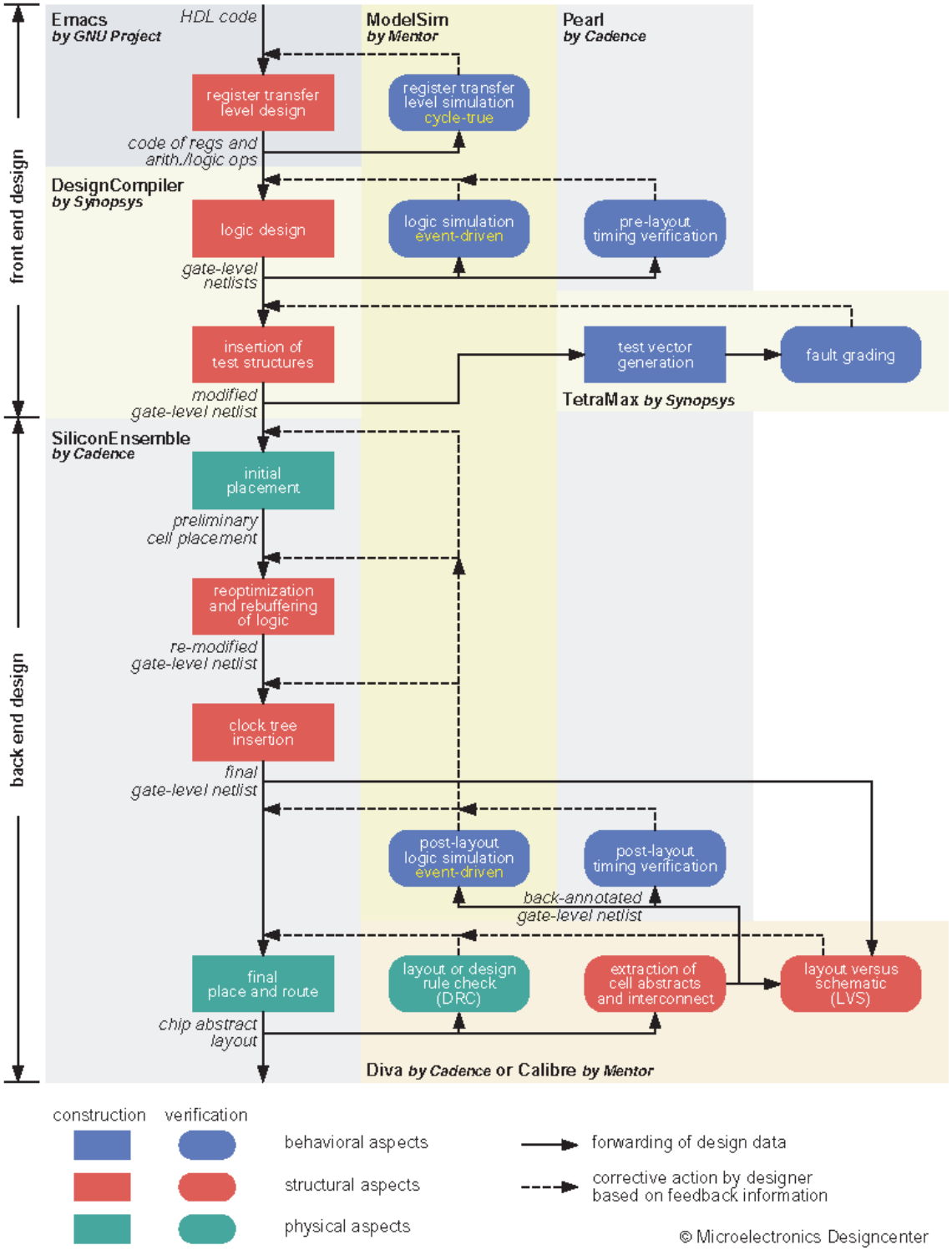



Figure 1.1: A typical digital VLSI design flow

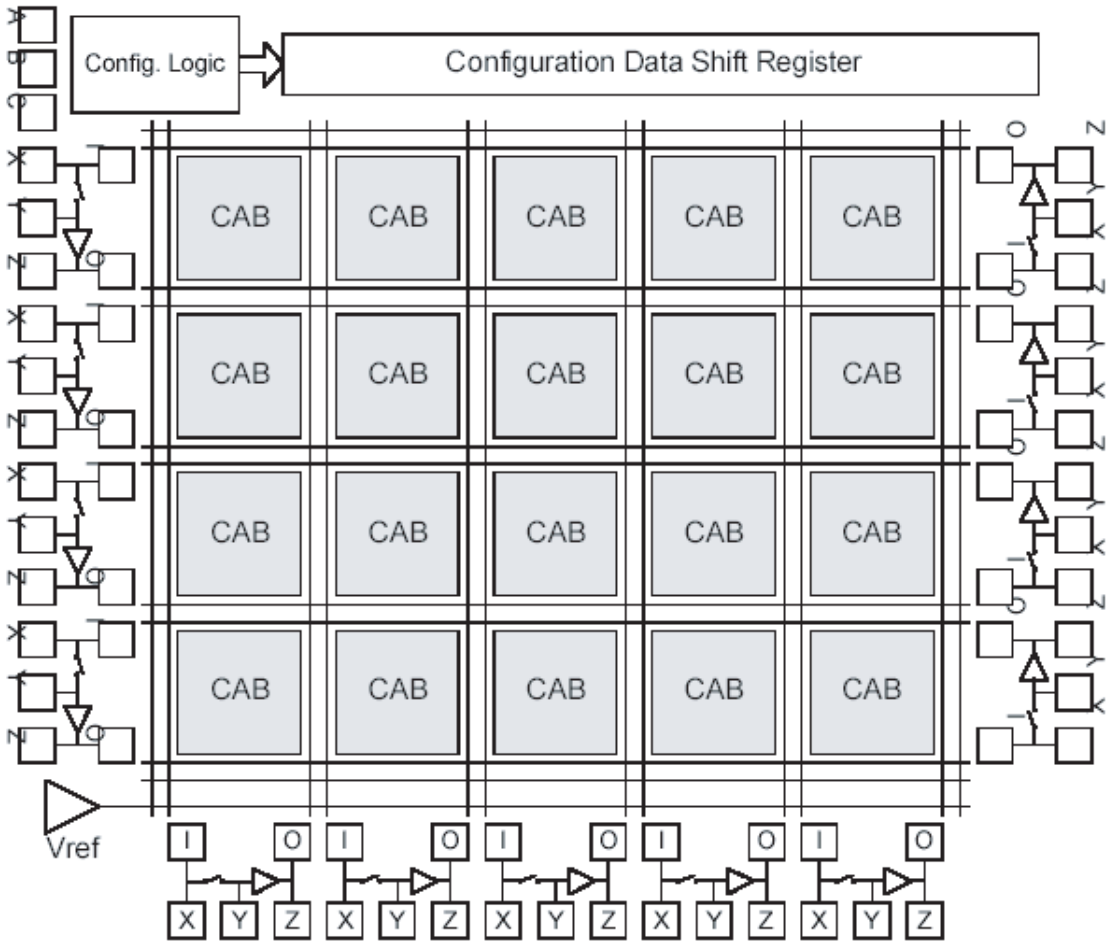



Figure 1.2: Anadigm's Field Programmable Analog Array AN10E40

## 1.3 Evolution of FPAA and Other Programmable Analog Devices

The first field-reconfigurable analog IC, originally intended primarily for synthesis and test of analog neural-network architectures, was proposed by Sivilotti [8]. CMOS transmission gates were used as the active switch elements that connected basic resources such as differential pairs and current mirrors in a hierarchical routing network. On board memory (SRAM) was provided for storing the state of each switch element but no memory was provided for storing circuit coefficients. In later work, Lee and Gulak [9] presented a low power FPAA based on MOS subthreshold circuit technique, where pass transistors controlled by SRAM based memory elements, were used as the active switches. Multi-valued memories were used to store circuit coefficients. However die-to-die variations in subthreshold model parameters brought challenges to circuit operation.

Simultaneously with [9] above, two patents were filed describing the design of an FPAA. Pilkington Micro-Electronics [10] described an array of operational amplifiers and associated programmable resistors and capacitors. Pass transistors were used as interconnect switches, while programmable resistors were constructed from multiple pairs of complementary MOS transistors. Each resistor was individually compensated to allow for manufacturing tolerances and temperature variations. Capacitors with value of 5e-12 Farad were fabricated, which were then multiplied 'by two impedance converters to final value of 5e-9 farad. Its applications were in the area of graphic equalizers, audio mixer desks, special purpose filters, spectrum analyzers, signal generators, prototyping, bands-free circuits for telephones,

and education. Sako [11] also described an FPAA design consisting of operational amplifiers, passive resistor and capacitor elements interconnected with pass transistors. More recently, Pankiewies et al. proposed a CMOS implementation of OTA based FPAA [12] which was especially attractive for analog filter applications.

Some commercial programmable analog ICs are also available. One of the first is GAP-01 [13]. This is the first attempt by industry to define a universal analog building block that could be used in several applications by externally routing signals present on the pins of the package. The first switched capacitor based FPAA was proposed by IMP [14] in 1995. It aims at general-purpose signal conditioning tasks in medical, industrial or other instrumentation and control systems, but the bandwidth is very small, only 150KHz at unit gain. This product was withdrawn from market in 1997. In the same year, Zetex [15] introduced the first continuous-time based analog programmable device - TracTM. The bandwidth increased to 4MHz., but the functionality it can realize is limited. By now probably the most successful FPAA products are from Anadigm (the former FPAA group of Motorola). Anadigm's FPAAs are also based on switched capacitor technique. The bandwidth of their products has increased from 250KHz (AN10E40) to 2MHz (AN20E40) [16]. A set of pre-designed analog module libraries (CAM - configurable analog module) and a software package are provided with Anadigm's FPAA chip. Many analog functions can be easily implemented with Anadigm's FPAA quickly.

## 1.4 Motivation of this work

There have been several programmable analog circuits available in the literature as well as some commercial chips available on the market. However, the functionalities they implement are relatively limited and their bandwidth is small. A general purpose FPAA with good supporting CAD tool suitable for high frequency applications has not yet appeared. From circuit design point of view, this could be due to (1) Most of previous designs are based on switched-capacitor technique, thus the system bandwidth is limited by the clock and sampling rate; (2) Many of them use MOS transistor based switches. When the array size grows, the numerous switches can contribute significant amount of parasitics into the circuit and dramatically degrade the system performance. In the area of design automation, very few papers [17], [18] available address the CAD tools development for analog arrays or other programmable analog devices. The difficulty mainly comes from the inherent difference between analog and digital systems in many aspects:

*(1) Loose form of hierarchy: the hierarchical decomposition of digital systems is clearly defined with well-accepted levels (Figure 1), while analog designs have a loose form of hierarchy because the hierarchical decomposition in analog is based on an intuitive structural decomposition of the modules, rather than the properties of signal type and corresponding time representation at different abstraction levels as in digital.*

*(2) Large spectrum of specifications: more performance specifications are imposed on analog circuits than the digital ones. In addition, the specifications often*

*impose conflicting requirements on the design. This results in many trade-offs to be managed during the design of the circuit, usually a multidimensional problem which is difficult to handle (Figure 3 [19]).*

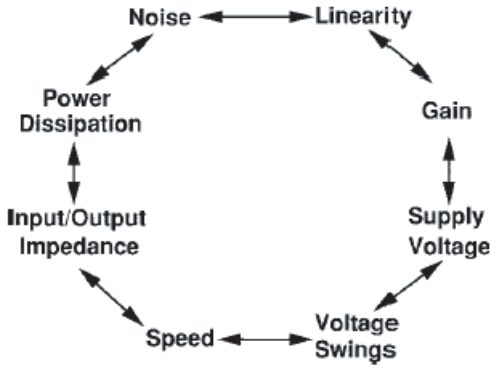



Figure 1.3: Analog Design Tradeoffs

*(3) Big influence of technology: technology and environmental parameters show a larger influence on analog circuits. Process, biasing or temperature variations and layout parasitics strongly influence the circuit performance and can even change the functionality of the circuit.*

*(4) Interactions at the system level: Analog circuits are also very sensitive to interactions at the system level. The interactions may be between two analog blocks, or between analog block and digital block of a large system such as clock noise. Similarly, if several different channels of a data-acquisition system are integrated on one chip, strong crosstalk may happen between these channels and cause serious signal integrity issue.*

(1) and (2) make automatic technology mapping and placement prohibitive to implement for FPAAs. Because of (3) and (4), the parasitics induced perfor-

mance degradation such as loading and coupling effect is much more complicated for FPAAs than that of FPGAs. To attack these difficulties, we need 1) flexible and efficient internal CAB and FPAA architecture; 2)rich IP portfolio which provide high performance, pre-qualified analog/mixed-signal IPs; 3)good supporting CAD tools. Apparently there are a lot of work involved. So this work is an attempt to provide some initial solutions for these areas with focus on analog IP design and FPAA router. Although the idea originates from the concept of a field programmable analog array, the author is trying to go beyond the array based approach and develop a hierarchical analog/mixed-signal design approach by taking advantage of the flexibility that laser Makelink provides. A hierarchical based design is configurable and suitable for CAD methodologies. It provides pre-qualified software and hardware components, and is able to translate complex analog circuits to a simple set of high-level functions. So it's ideal for building prototype systems or low volume analog ASICs for it's quick-to-market time.

It should also be noted that in this array based FPAA architecture, there are abundant interconnect routing resources. The coupling between the wires and the noise from substrate may be a serious issue. Therefore, careful layout design is extremely important. In this work, common centroid, interdigitated device structures, and dummy devices are used extensively to improve matching.

Laser Makelink is an essential programming technology in this work. Makelink's are not only used as routing switches, but also used as a trimming method to improve the precision and reduce the cost due to the extra circuits.

### 1.4.1 TSMC018 CL/CM Process

Most of the designs in this work were done on TSMC 0.18um Mixed-Signal Mode Process with 3.3 $V$ power supply. The default features of this Mixed-Signal process include: $1p6M$, 1.8 $V$/3.3 $V$ MOS transistors, deep N-Well, linear MIM capacitor, spiral inductor, MOS varactor, junction varactor, poly/diffusion resistors and thick top metal interconnect [20].

### 1.4.2 Potential FPAA Applications

FPAAs and the hierarchical designs won't be suitable for large volume semiconductor analog products, such as in the sectors of flat panel display, storage, consumer electronics ... However, it's a cost efficient solution for a relatively small volume, analog ASICs or for quick system prototyping or verifications. The potential applications include:

- Signal Amplification, Summation, Filtering, Integration
- Signal Conditioning for A/D Converters: buffer, pre-amplifier
- Flexible AFEs for Data Acquisition
- Industry and Aerospace Control Circuit Block (PID application)
- Sensor Signal Conditioning
- Precision Voltage Monitoring

They may be used in the areas include discrete PCB design integration, aerospace applications which requires radiation-hard design, or as a sub-system of an SoC or structured ASIC.

# Chapter 2

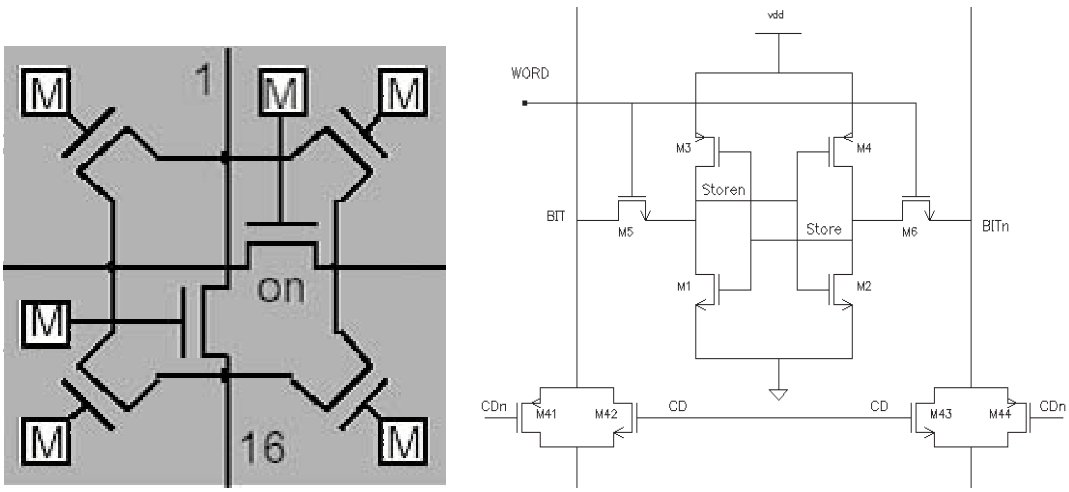
# Programming Technology

## 2.1 Programming Technology Overview

FPGAs/FPAAs can be categorized by how they are being programmed, i.e., how the switches are implemented. The programming technology has critical impact on the system performance. The existing options today include SRAM controlled MOSFET switch, antifuses and EPROM/EEPROM.

### 2.1.1 SRAM

SRAM controlled MOSFET switch (or transmission gate) is probably the most widely used programming technology [22]. An SRAM-based FPGA/FPAA is programmed by loading the configuration bit stream from an external source into the on-chip SRAM memory. Each switch, in most cases an MOS transistor, in the CAB/logic and routing interconnect is controlled by a memory cell. Figure 2.1 is a typical switch matrix and the controlling SRAM cell. Using SRAM programming technology, users may reuse chip during prototyping to reduce cost, and a system can be configured using ISP (in system programming). SRAM programming is also useful for upgrade - manufacturer may send customers a new configuration file




(a) An SRAM controlled switch matrix. M represents an SRAM cell

(b) A 6-transistor SRAM cell [21]

Figure 2.1: SRAM controlled MOSFET switch

instead of a new chip to upgrade the system function. However, SRAM's biggest advantage "reconfigurability" also brings a disadvantage - volatility. When the power is off, the configuration data is lost. So an SRAM based FPGA/FPAA must be reprogrammed each time power is applied. And SRAM based programmable devices often cost more silicon area. Moreover,the relatively high resistance of the MOS switch may severely limits the overall system bandwidth. As shown in figure 2.2, each terminal of the MOSFET switch is associated with some parasitic capacitance. As the number of switches grows, those parasitic capacitors and resistors will dramatically slow down the speed.

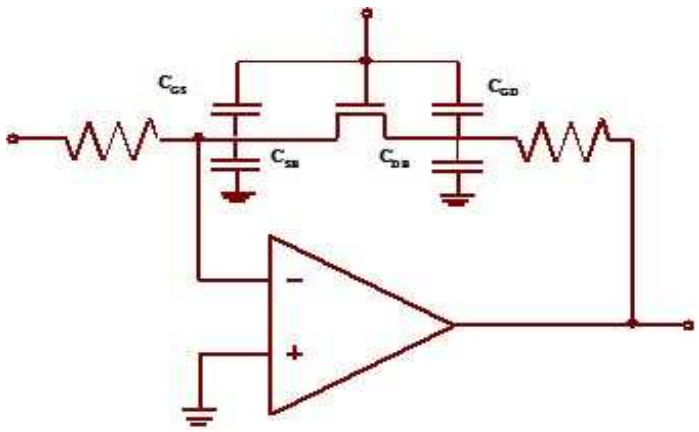


Figure 2.2: The parasitic capacitance associated with an MOSFET switch

### 2.1.2 Antifuse

An antifuse is the opposite of a regular fuse - an antifuse is normally an open circuit until a programming current flowing through it.

#### Actel's Antifuse Technologies

Actel's oxide-nitride-oxide (ONO) antifuse is a well known programming technology. In this poly-diffusion antifuse, the high current density causes a large power dissipation in a small area, which melts a thin insulating dielectric between polysilicon and diffusion electrodes and forms a thin, permanent, and resistive silicon link. The programming process also drives dopant atoms from the poly and diffusion electrodes into the link, and the final level of doping determines the resistance value of the link. Actel calls this antifuse a programmable low-impedance circuit element (PLICE) [23]. Figure 2.2 shows a poly-diffusion antifuse with an ONO dielectric sandwich of $SiO_2$ grown over the n-type antifuse diffusion, a $Si_3N_4$ layer, and another thin $SiO_2$ layer [24]. The average resistance of a blown antifuse are controlled

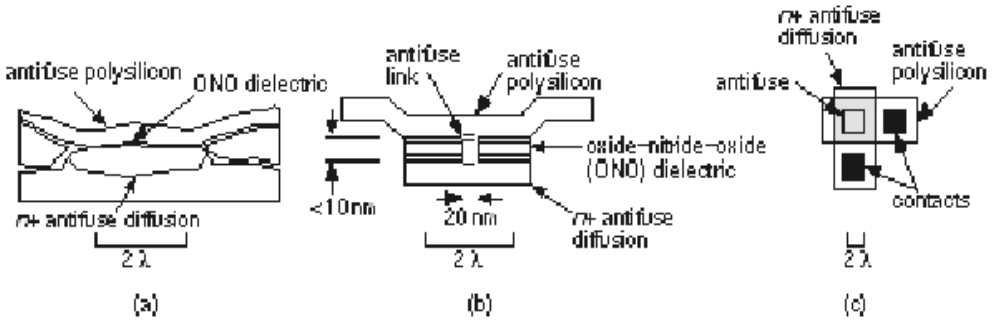



Figure 2.3: Actel antifuse (a) A cross section; (b) A simplified drawing (c) top view

by the fabrication process and the programming current, but actual values may vary in the range between 100 - 800 Ω with nominal value of about 500 Ω. ONO antifuse has smaller footprint and it's radiation tolerant, but its fabrication requires modifications of the standard CMOS process. For examples, a double-metal, single-poly CMOS process typically uses about 12 masks-the Actel process requires an additional three masks. The n-type antifuse diffusion and antifuse polysilicon require an extra two masks and a 40 $nm$ (thicker than normal) gate oxide (for the high-voltage transistors that handle the programming voltage) uses one more masking step. And it's a weak one dimensional filament, which is not suitable for carrying high current.

Actel also has another antifuse called M2M. M2M antifuse is composed of layers of amorphous silicon and dielectrics, sandwiched between top metal and the via-plug that is used for connecting lower metal to the top metal. Application of a 15V programming pulse causes a phase change within the amorphous silicon. A filament of crystalline silicon forms between the metal layers. That filament is a mixture of silicon and the metal-layer material. Typical connection resistance is 20 Ω to 100 Ω.

## Quicklogic Metal-Metal Antifuse Technology

Figure 2.4 shows a QuickLogic metal-metal antifuse (ViaLink$^{TM}$). QuickLogic ViaLink is a Tungsten plug connecting the two metal layers with a layer of amorphous silicon antifuse material deposited on top. The amorphous silicon provides a high resistance layer (>1 $G\Omega$) insulating the Tungsten plug. When the programming voltage is applied the amorphous silicon is converted to low resistance silicon with resistance of typically 80 $\Omega$.

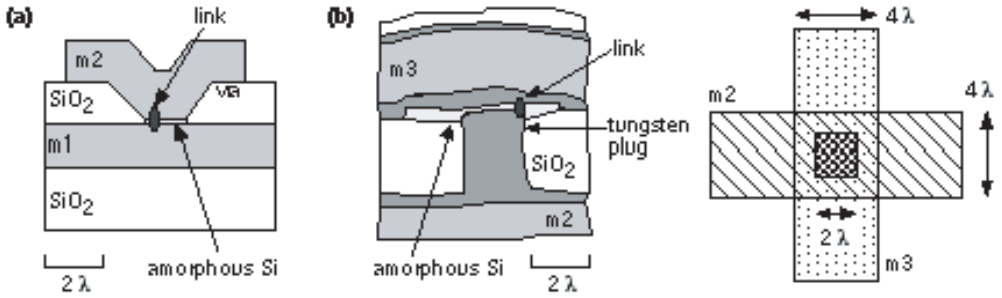



Figure 2.4: Metal-metal antifuse. (a) An idealized cross section of a QuickLogic metal-metal antifuse in a two-level metal process. (b) A metal-metal antifuse in a three-level metal process that uses contact plugs. The conductive link usually forms at the corner of the via where the electric field is highest during programming.

There are two advantages of a metal-metal antifuse over a poly-diffusion antifuse. The first is that connections to a metal-metal antifuse are direct to metal-the wiring layers. Connections from a poly-diffusion antifuse to the wiring layers require extra space and create additional parasitic capacitance. The second advantage is that the direct connection to the low-resistance metal layers makes it easier to

use larger programming currents to reduce the antifuse resistance. The nominal QuickLogic metal-metal antifuse resistance is approximately 80 Ω (with a standard deviation of about 10 Ω) using a programming current of $15mA$ as opposed to an average antifuse resistance of 500 Ω for a poly-diffusion antifuse.

The size of an antifuse is limited by the resolution of the lithography equipment used to makes ICs. The Actel antifuse connects diffusion and polysilicon, and both of these materials are too resistive for use as signal interconnects. To connect the antifuse to the metal layers requires contacts that take up more space than the antifuse itself, reducing the advantage of the small antifuse size.

An antifuse is resistive and the addition of contacts adds parasitic capacitance. The intrinsic parasitic capacitance of an antifuse is small, but to this we must add the extrinsic parasitic capacitance that includes the capacitance of the diffusion and poly electrodes (in a poly-diffusion antifuse) and connecting metal wires. These unwanted parasitic elements could add considerable RC interconnect delay if the number of antifuses connected in series is not kept to minimum. Clever routing techniques are therefore crucial to antifuse-based FPGAs [22]. The long-term reliability of antifuses is an important issue. , Actel's research has shown that the programmed link is fragile under over-current conditions. Such conditions occur frequently in normal operation, making the field reliability of amorphous antifuses questionable. High circuit speeds and large array sizes increase the likelihood of over-current failure, limiting the speed and size attainable with an amorphous antifuse. The programmed antifuses sometimes revert to a high-impedance state due to cracking or a phenomenon called read disturb. The result is that the antifuse's

resistance jumps, which will change the corresponding logic circuit's propagation delays and may even look to the logic like an open-circuit. This reversion tends to be self-healing; normal logic-high voltages are sufficient to reprogram the disturbed antifuse. However, there is no guarantee that the node containing the disturbed antifuse will see a logic-high voltage again, once the change has occurred. Thus, the tendency to self-heal is not a reliable antidote. Therefore, the designer using the Actel M2M or QuickLogic antifuse must limit the current flow through them to avoid stressing the filament, and it's virtually impractical to use it for analog design

### 2.1.3 EPROM and EEPROM

UV-erasable electrically programmable read-only memory (EPROM) cells are used in many programmable devices such as Altera MAX 5000 EPLDs and Xilinx EPLDs as their programming technology. Altera's EPROM cell is shown in Figure 2.5 [24]. The EPROM cell is almost as small as an antifuse. An EPROM transistor looks like a normal MOS transistor except it has a second, floating gate (gate1 in Figure 2.5). Applying a programming voltage VPP (usually greater than 12V) to the drain of the n- channel EPROM transistor programs the EPROM cell. A high electric field causes electrons flowing toward the drain to move so fast they "jump" across the insulating gate oxide where they are trapped on the bottom, floating gate. We say these energetic electrons are hot and the effect is known as hot-electron injection or avalanche injection. EPROM technology is sometimes called floating-gate avalanche MOS (FAMOS).

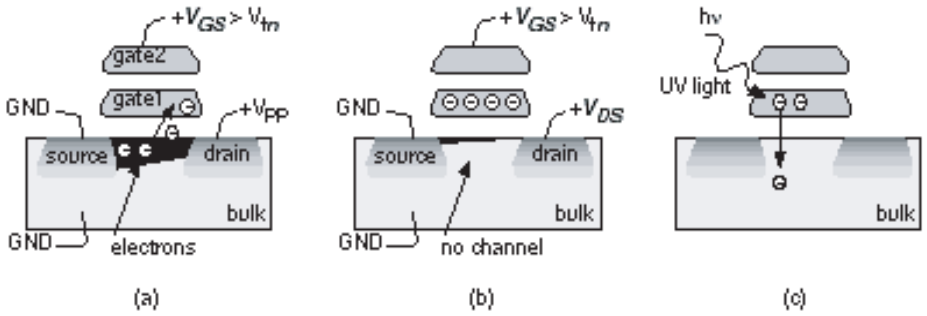



Figure 2.5: An EPROM transistor. (a) With a high (> 12 V) programming voltage, V PP , applied to the drain, electrons gain enough energy to "jump" onto the floating gate (gate1). (b) Electrons stuck on gate1 raise the threshold voltage so that the transistor is always off for normal operating voltages. (c) Ultraviolet light provides enough energy for the electrons stuck on gate1 to "jump" back to the bulk, allowing the transistor to operate normally.

Electrons trapped on the floating gate raise the threshold voltage of the n-channel EPROM transistor. Once programmed, an n-channel EPROM device remains off even with VDD applied to the top gate. An unprogrammed n- channel device will turn on as normal with a top-gate voltage of VDD. The programming voltage is applied either from a special programming box or by using on-chip charge pumps. Exposure to an ultraviolet (UV) lamp will erase the EPROM cell. An absorbed light quantum gives an electron enough energy to jump from the floating gate. The manufacturer provides a software program that checks to see if a part is erased. EPLD parts are available in a windowed package for development, erase it, and use it again, or in a nonwindowed package and program (or burn) the part once only for production. The packages get hot while they are being erased, so that

windowed option is available with only ceramic packages, which are more expensive than plastic packages.

Programming an EEPROM transistor is similar to programming an UV-erasable EPROM transistor, but the erase mechanism is different. In an EEPROM transistor an electric field is also used to remove electrons from the floating gate of a programmed transistor. This is faster than using a UV lamp and the chip does not have to be removed from the system.

Advantages of EPROM/EEROM are their reconfigurability and non-volatility. But they have large resistance, occupy more silicon area and require multiple voltage sources to be programmed. Moreover, their fabrication is not compatible with standard CMOS processes.

## 2.2 Laser Makelink Technology

All of the programming technologies introduced above suffer from various problems, such as high resistance, large parasitic capacitance, incapable to carry large current and incompatible with standard CMOS processes. Ideally we wish the programmable switches have the properties of a metal wire. Thus laser Makelink technology is the most promising candidate, especially for analog/mixed-signal applications.

Laser processing techniques have been used in semiconductor industry for many years. Laser-induced cutting was one of the successful examples. The technology was first commercialized by IBM in 1979 [25], in which a laser with a 1060 $nm$

wavelength was used to cut off the defective memory cells and "replace" them with redundancies. During the early years, poly-silicon was the target material. However, with the development of multi-level metallization, deeply buried polysilicon lines have become harder to cut. Laser diffused link was also reported, but high resistance and current leakage limited its commercial application [26]. Hence, people started looking at the shallower metal layers. Open-window metal cuts have been found in commercial devices, like LPGAs, but the exposed metal can evoke reliability concerns and the process requires extra-mask and process steps. The most favorable metal cut structure would be hermetic, no need for extra-mask and compatible with the standard CMOS process. Unfortunately, recent study indicates that the applicable laser processing window for buried cuts is too narrow to satisfy the yield [27].

As a complementary scheme, laser-induced metal antifuse, i.e., laser Makelink, has been proposed [28] [29] [30] [31] which has shown much broader process window and higher yield. The electrical connection is formed vertically between two levels of metallic interconnects by applying an IR laser pulse (1047 nm wavelength) with a time frame of several nanoseconds. This link structure possesses inherit advantages: extremely low parasitics, strong connections, high reliability hermeticity, radiation hardness, and CMOS process compatibility. Thus, this kind of link can be widely implemented in digital logic and analog circuit integration.

### 2.2.1 Laser Makelink Principle

Figure 2.6 is the schematic of a typical vertical Makelink structure. A laser Makelink is an electrical connection formed between two layers (vertical link) or within the same layer (lateral link) of metallization by a commercial pulsed IR laser. The principle of link formation employs the contrast of material properties between

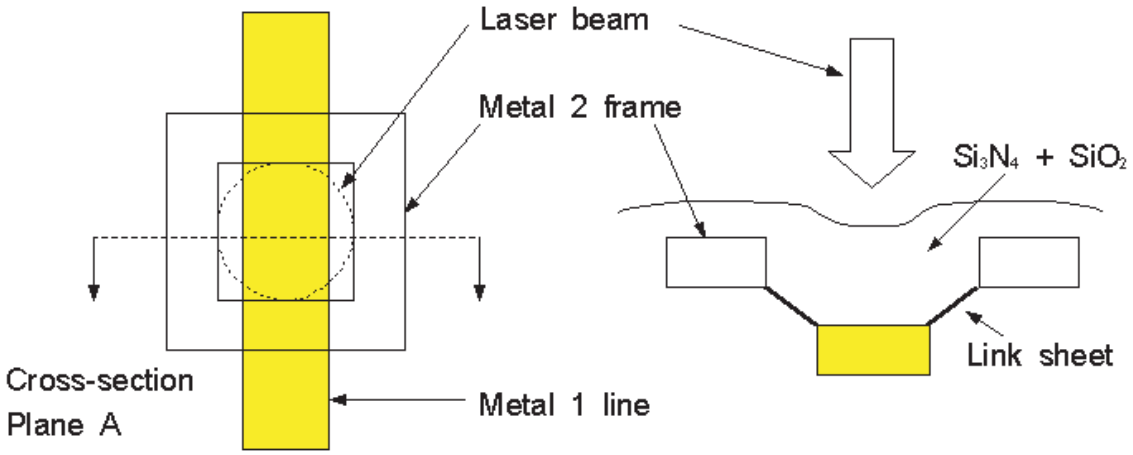
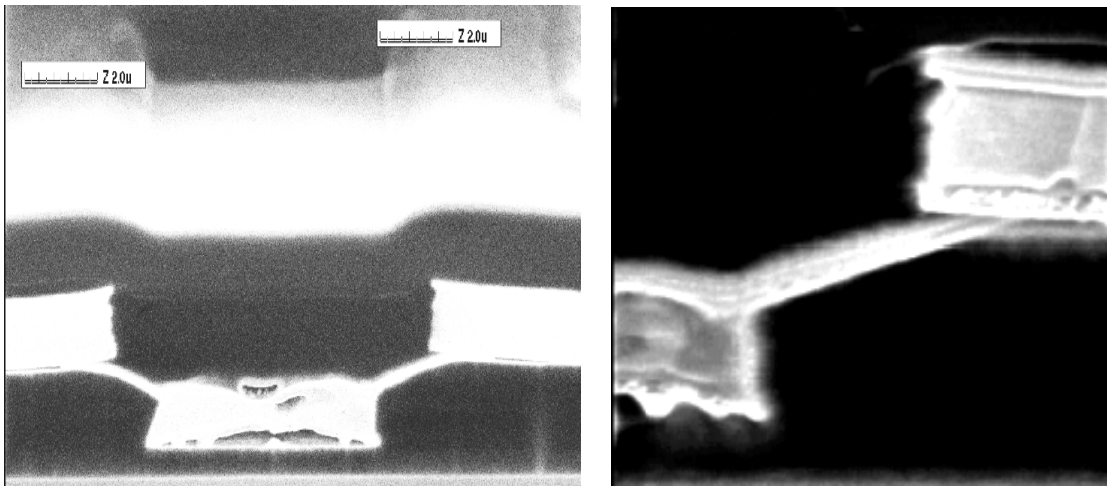



Figure 2.6: Vertical Laser Makelink structure (a) top view (b) cross-section view

the metal and the surrounding dielectrics $SiO_2/Si_3N_4$. The IR laser beam passing through the square hole of the upper metal (M2) frame is impinged on the lower metal (M1) line with negligible loss of energy in the covering dielectrics. The laser energy is absorbed on the surface of M1 to be resulting in a sharp metal temperature increase. Due to the extremely low thermal conductivity and light absorbency of

the dielectrics, the dielectric temperature is not changed so much. In the mean time, metal expansion fractures the surrounding dielectrics along the stress concentration paths and molten metal fills in the crack. At an optimal laser energy and spot size, dielectric cracks can be controlled to initiate from the upper corners of the M1 line and terminate near the inside lower corners of the M2 frame without propagating to the outside of the structure or fracturing the top dielectric passivation. An FIB cross-section image of a laser-induced Makelink interconnecting structure is shown in the following figure.




(a) (b)

Figure 2.7: FIB cross-section of a vertical Makelink structure

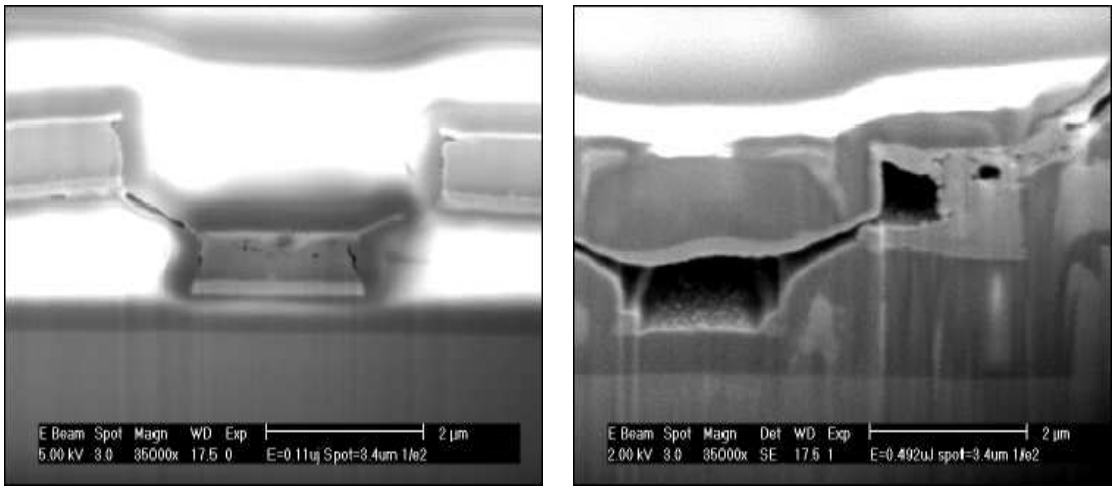
### 2.2.2 Laser Makelink Design

Much effort has been made to investigate the laser-metal interaction and so-induced thermal/mechanical phenomena in different link structures[35]. Both experiments and simulations indicated a broad processing window in term of laser

energy that ensures a good tolerance of laser errors as well as the fabrication-caused variation. There are several factors involved in the successful laser linking process. Generally, those factors can be divided into three categories: process characteristics, laser conditions and link geometry structure. In most cases, designers have little control on the process parameters. So only the latter two are discussed here.

- Energy Effect: Laser conditions include single pulse energy, pulse duration, shape and laser spot size. Among them, only laser energy and spot size are adjustable. In most cases, spot size is determined by the geometry size of the link structure and is usually a fixed parameter. The choice of the laser energy depends on the specific process parameters (such as dielectric material and thickness etc) and the link structure. Due to the non-uniformity of the temperature distribution, the thermal stress-induced crack initiation time and propagation direction are different around the annulus. Heat conduction along the metal line causes a fairly deep temperature gradient beyond the thermal diffusion length in a single pulse duration. If the energy is too low, the cracking will stop in the middle between two metal lines and fail to form a valid electrical connection (figure 2.8 (a)). If the energy is too high, the crack will continue to propagate along the bottom plane after it reaches the frame. Excessive metal flow results in large voids in the lower metal that increases electro migration risk; in the mean time, the undesired crack outside the link frame can destroy the completeness of the top passivation (figure 2.8 (b)).

  In order to characterize the yield and robustness of a Makelink structure, it's




(a) (b)

Figure 2.8: Energy effect on the vertical link (2um bottom metal line, 4um hole) formation (a)$E = 0.11uJ$; (b)$E = 0.49uJ$

useful to define an appropriate laser process window. The process window in term of absolute energy lacks universal significance. Thus a normalized window is preferred[32].

$$RelativeEnergyWindow = \frac{E_H - E_L}{E_{Avg}} \tag{2.1}$$

where $E_H$, $E_L$ and $E_{Avg}$ are the high, low and average energies, of which a link can be formed, respectively. The relative window is a normalized, non-dimensional term that eliminates the dependence of the absolute energy window on the characteristics of different laser systems. It has been shown that an acceptable energy window will always be found for the metal link process for aluminum metallization processes insulated by $SiO_2$ dielectric[33].

- Geometry Effect: Zero gap is desired in order to increase the link density.

However, for a laser beam with Gaussian energy distribution, a link structure with zero-gap is not an efficient design. For example, for a link structure with a 4um line, a 4um hole with zero gap and a 2um frame width, 38.4% laser energy is absorbed by the frame, if the FWHM laser spotdiameter is equal to the metal line width[34]. Thus, the available energy window is significantly reduced due to the increased probability of frame damage. Besides, due to the lens effect of the passivation over the metal2, a part of the laser energy could be absorbed by the frame face inside the hole. The lower corner of the metal2 frame is heated up more quickly than for a planar structure receiving normal incident laser bean. This lens effect causes undesired link formation from the upper corner[34].

Based on our extensive simulation and experiment results [35], we came up a basic rule of thumb: for vertical Makelink, the horizontal gap between the top and bottom level of metal should be roughly equal to the thickness of the dielectric, because the vertical link usually forms in the 45° direction; for later link, the distance between the two adjacent metal wires is set to be equal to the size determined by the specific design rule.

Vertical laser Makelink's have been successfully demonstrated on various CMOS processes with aluminum metallization [35]. The successful, low resistance link formation and link yield are highly dependent on the specific process. Sometimes, later link structures may provide better results. For examples, figure 2.9 shows four later link structures designed on National Semiconductor's 0.18 $um$, five-layer

metallization process. Vertical link designs have been proved to be unsuccessful for this process. Figure 2.10 shows the energy window of the four structures and their average resistance per link with standard deviation smaller than 1 $\Omega$ ((pitch 2.2 $um$ test chip). Structure 4 has the lowest average resistance and standard deviation, but it also has the narrowest energy window, because the links were laid out on the top metal layer where a small energy increase may easily break the passivation layer $Si_3N_4$. Structure 3, which shows the highest average resistance and standard deviation within the window, indicates that the three-line design in metal 4 layer has high resistance with a large variation, but it is likely to increase the probability of link formation. For structure 2, 3 and 4, an optimal energy exists within the energy window which produce the smallest resistance. Figure 2.11 is the link yield for different structures. For this specific CMOS process, structure 2 achieves highest yield and lowest resistance per link simultaneously at the optimal energy 0.25 $uJ$. Furthermore, its energy window curve follows its yield curve. The experiment results show that the optimal energy for structure 2 and 3 are 0.25 $uJ$ and 0.22 $uJ$, respectively. In the case of structure 4, with 0.25 $uJ$ energy, 100% yield was obtained. (no test chain's open or short) at the cost of a slightly higher average resistance[36]. Laser Makelink's are not necessary limited to aluminum links. They can also be formed using copper. Some novel copper test structures are now being developed on IBM's BiCMOS SiGe process in Peckerar & Bernstein's group. Appendix A shows some laser Makelink test chips designed on various CMOS processes and the IBM SiGe Copper process.

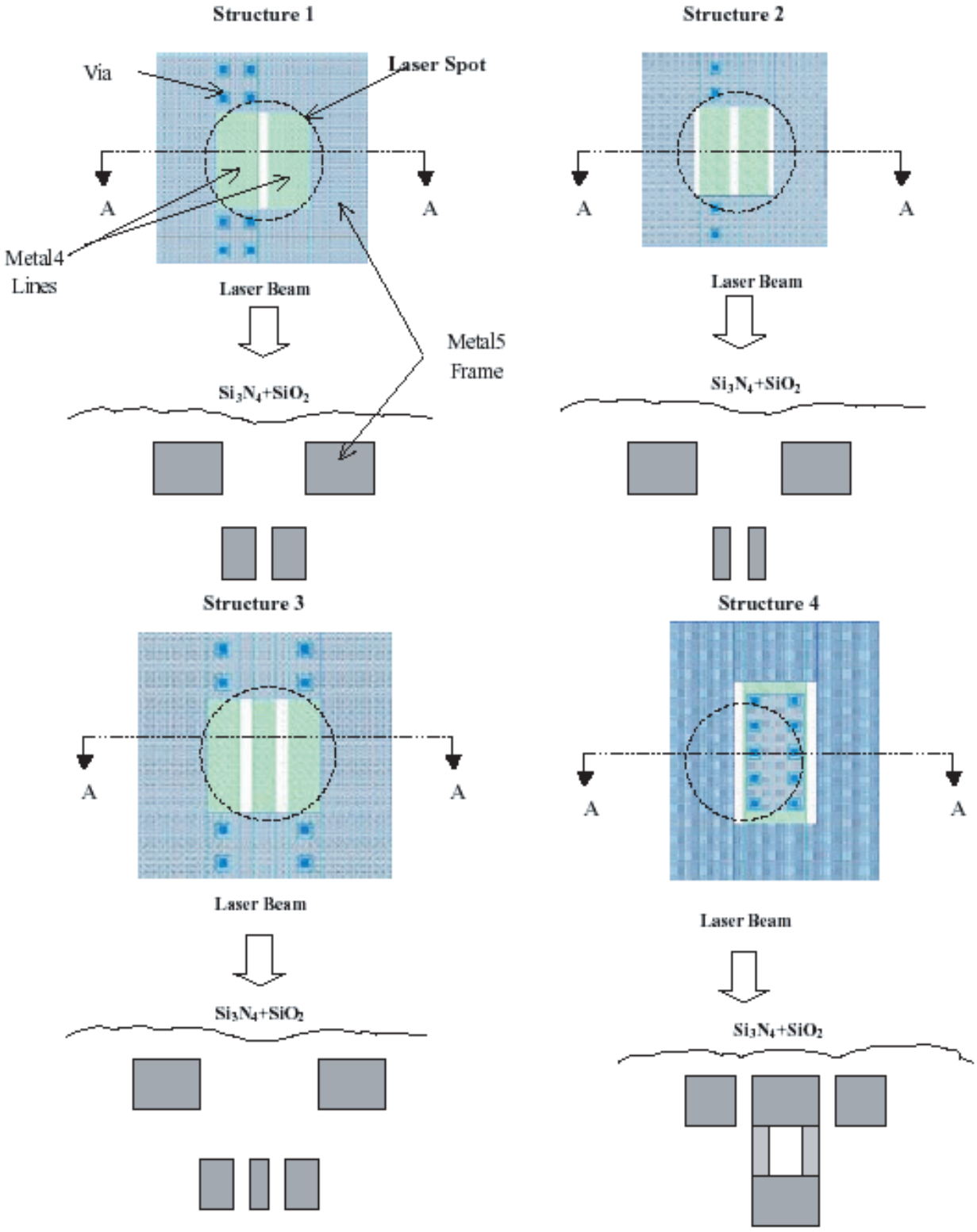



Figure 2.9: Four later link structures design for NSC's 0.18 *um* CMOS process

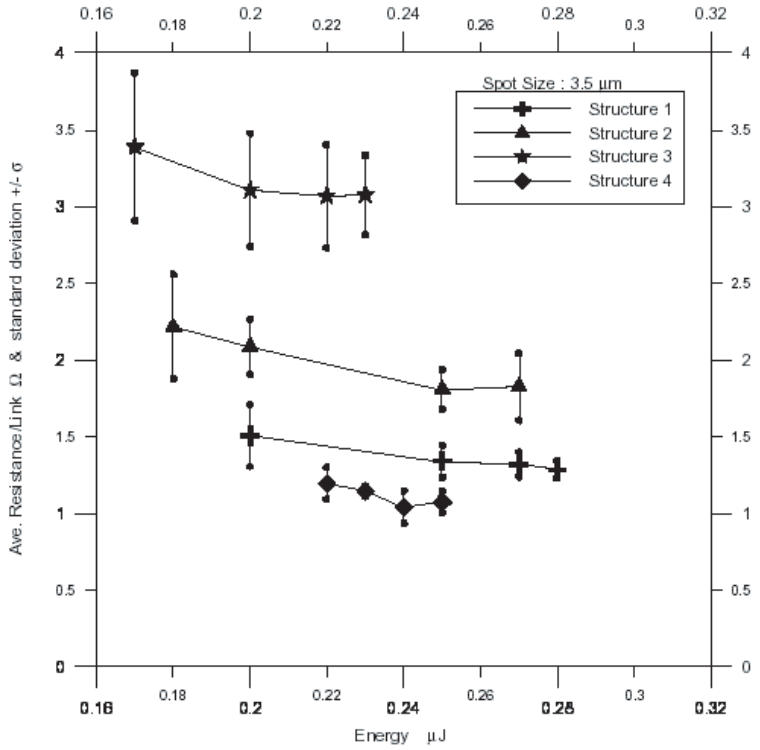



Figure 2.10: Energy windows of the four later link structures and their average resistance per link (2.2 um pitch)

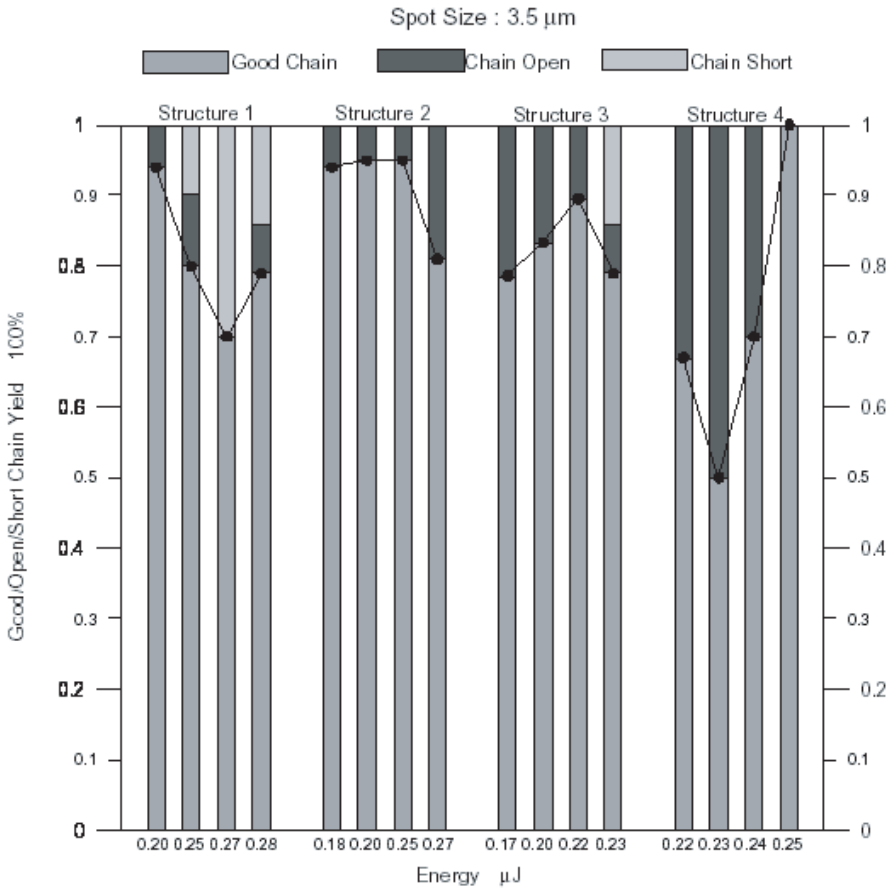



Figure 2.11: Test chain yield of the four later link structures

### 2.2.3 Summary

Advantages of Laser Makelink in the application of programmable devices include:

- Makelink is the ideal programmable switch; among all current programming technologies, Makelink offers the lowest programmed switch resistance and unprogrammed switch capacitance. For example, a typical Makelink switch resistance is approximately 1 $\Omega$, which is about 2-3 orders smaller than that of Actel antifuse or a MOS switch. This makes the laser Makelink technology ideal for high speed, low power and low noise FPAA applications.

- High reliability and tolerant to high current density: n analog application, the switches are required to be able to carry large current. Therefore, ONO and M2M antifuses cannot be used.

- Leakage Current: Because there are many antifuses on a chip, leakage currents can amount to considerable power consumption. A 10 $nA$ leakage current in each of the typical 750,000 antifuses on a large FPAA would waste 7.5 $mA$. Since the amorphous silicon/dielectric layer in ONO and M2M antifuses are very thin, they produce significantly higher leakage current than does Makelink.

- Area efficiency: Compared with SRAM technology, Makelink can save significant amount of silicon. For example, on a commercial CMOS process, the minimum width transistor area can be represented by MWTA. For a SRAM

controlled MOS switch, the number of MWTAs needed for a switch is 1+5 = 6 (assume a five transistor SRAM cell). While for laser Makelink based FPAA, no SRAM cell is needed to control the laser Makelink switch. In fact, only top 2 levels of metals are used and no silicon area is occupied. The silicon under the routing interconnects could be used to build more active devices and larger passive element matrices. Furthermore, considering its radiation hardness, Makelink will save much more area than traditional SRAM based technology. It is also worth noting that, at first glance, Makelink appears to occupy greater area than ONO and M2M antifuses. However, in fact, ONO and M2M antifuses require contacts to connect to the metal layers and these take up more space than the antifuse itself. Accordingly, ONO and M2M do not offer density advantages to Makelink; the contact and metal spacing design rules limit how closely the antifuses may be packed rather than the size of the antifuse itself.)

- CMOS compatible processing steps: Unlike ONO and M2M antifuse technology, Makelink is completely compatible with any commercial CMOS processes. No extra process step or photomask is required.

- Radiation Hardness: Since no active devices in the Makelink switch, it's inherently a radiation hard technology; Makelink consists of pure aluminum and is therefore truly radiation hard. Accordingly, Makelink-based LFPAA provide significant cost savings and are ideal for high-reliability space missions.

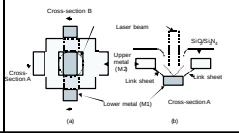
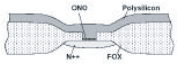
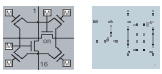
| | MakeLink™ | Actel M2M | Actel ONO | SRAM |
|---|---|---|---|---|
| |  |  |  |  |
| Radiation Tolerance | Strategic Radiation Hardness | Tactical Radiation Hardness TID = 500Krad(Si) | Tactical Radiation Hardness TID = 500Krad(Si) | Radiation Soft |
| Robustness | 2-Dimensional strong metal sheet (cross-section area about 0.5 $\mu m^2$) | 1-Dimensional weak metal silicide filament (cross-section area about 0.06 $\mu m^2$) | 1-Dimensional poly/$n^+$ diffusion filament | Robust, also Reprogrammable |
| Current Carrying Capacity | ~mA | ~$\mu$A | ~10$\mu$A | ~ $\mu$A-mA |
| Application | Analog, Digital and Mixed-signal | Digital | Digital | Analog, Digital, and Mixed-signal |
| Typical unprogrammed Capacitance | < 0.05fF | 2.9 fF | 7.7 fF | ~5-10 fF |
| Typical programmed Resistance | 1-10$\Omega$ | 25$\Omega$ | 125$\Omega$ | 100-1000$\Omega$ |
| Leakage Current | ZERO | 3 nA | 10 nA | ~nA |
| Net Link Area | 2X | 1X | 1X | 5X-6X |
| CMOS Compatibility | No additional mask levels | Additional mask levels required | Additional mask levels required | No additional mask levels |

Figure 2.12: Table 2.1 comparison between laser Makelink with other programming technologies.

## 2.3 Laser Makelink Applications

Laser Makelink is a metallurgic connection. It is similar to a via but much stronger, reliable and capable to carry high current density. Unlike SRAM based programming technology, where MOSFET just functions as a switch to route signal. Makelinks can be used in the core circuit blocks as a "mask metal lines". To the circuit designs, the beauty of laser Makelink is that it can give them the capability to reconfigure the circuit topology at almost equivalent to mask level even after fabrication. Moreover, laser Makelink can also be used as a low cost "trimming" method [37]. Redundant transistors, resistors and capacitors may be added along with the specific devices. Whenever it's necessary, the component value such as transistor aspect ratio W/L or resistance can be fine tuned for precision by adding/removing some redundant component(s). Figure 12 is an example of changing MOSFET W/L using Makelink. The detailed application of laser Makelink in active circuits will be discussed in the following chapters.

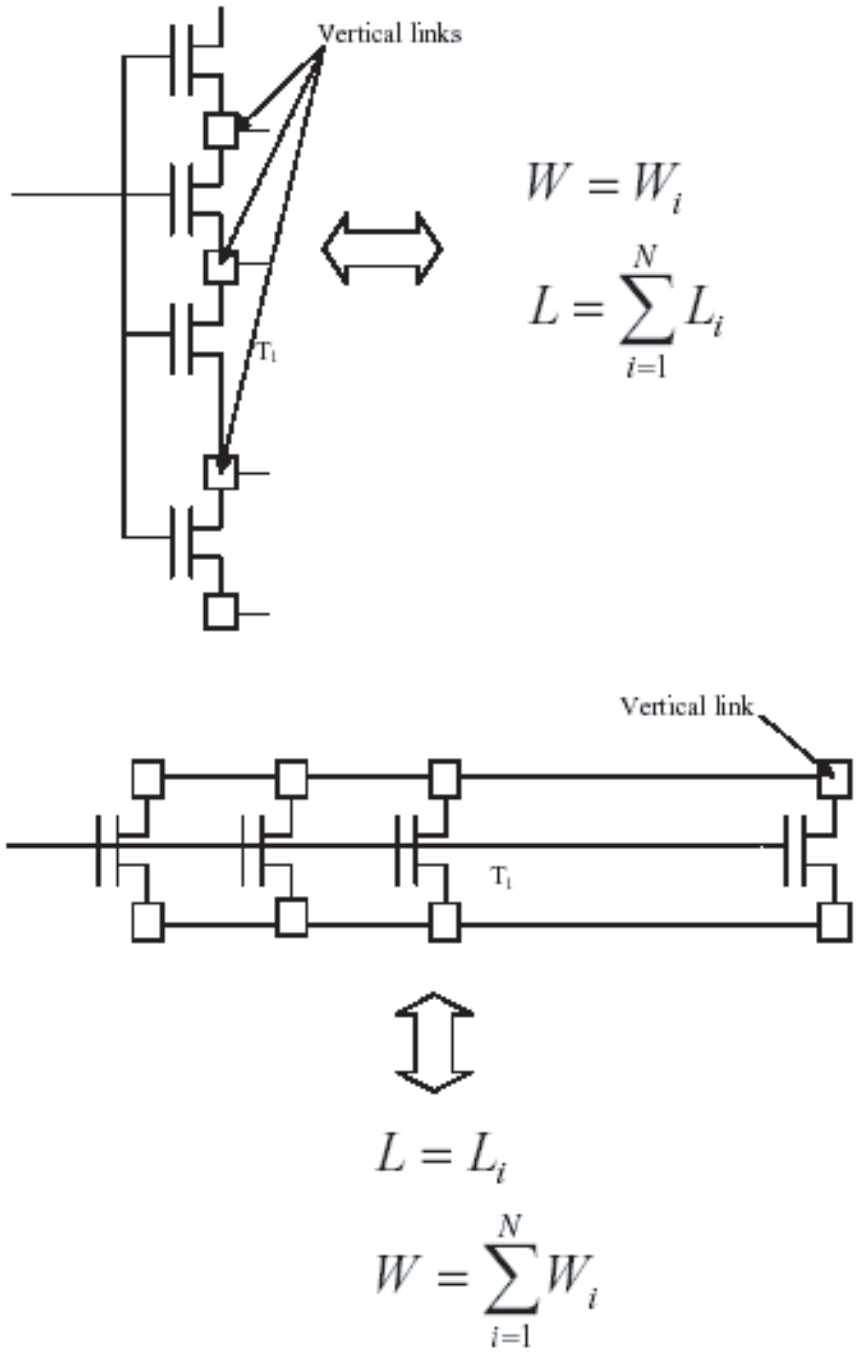



Figure 2.13: Reconfigurable MOS transistor aspect ratio

# Chapter 3

# Routing for FPAA

Similar to the FPGA, implementing an analog circuit on an FPAA requires a large number of switches to be programmed to the proper state so that the desired circuit topology and signal path can be established. Clearly, if the end user has

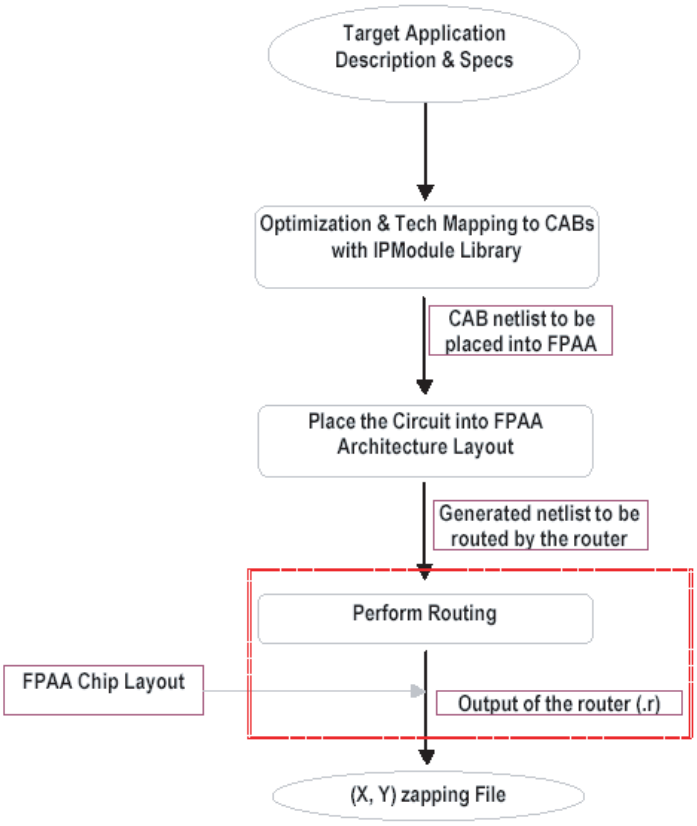



Figure 3.1: A simplified FPAA CAD design flow

to specify the state of each switch in the FPAA, the design cycle will be too long.

Therefore, the FPAA is designed so that the end user only describes a targeted application at a high level of abstraction, typically using a schematic entry with the IPmodule/CAM (configurable analog module) library provided by the manufacturer. Then, this high-level description is mapped and placed into a specific FPAA architecture. A netlist file, which describes a set of connections to be made, is generated after the placement phase. Then FPAA router takes this netlist file as input and performs routing. Combined with the chip layout, the end user will know which switches need to be turned on (e.g., laser programmed).

## 3.1 What is routing

The FPAA routing problem is defined as follows: Given a netlist and a placement of the CABs and IO cells, to route all the nets on the given FPAA architecture without exceeding the total available routing resources and without overly degrading the performance of the circuit [38].

Unlike custom analog IC designs, routing resources in FPAAs are fixed and limited. All connections must be completed within the horizontal and vertical channels, via Manhattan paths. The FPAA routing architecture not only affects routing but also has significant impact on the performance of the implemented circuit. To facilitate the FPAA router development, an array-based FPAA architecture was developed, as shown in figure 3.2.

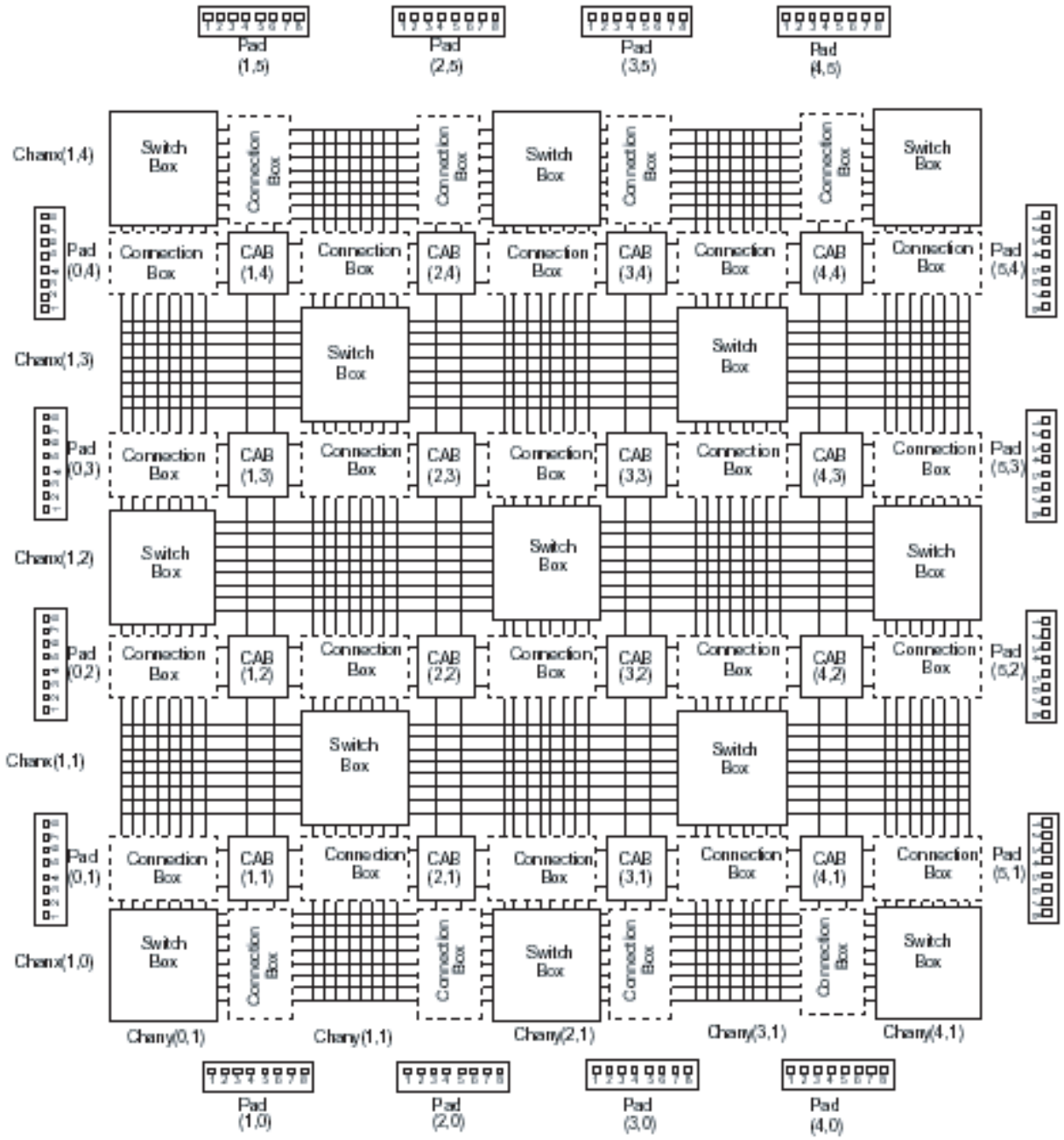



Figure 3.2: An array based FPAA architecture

### 3.1.1 Architecture Overview

The following notations were used to describe some important parameters of the FPAA routing architecture [40]. The number of wires or tracks contained in a channel is denoted by W, i.e., width of the channel. The number of wires in each channel to which a CAB pin can connect is called the connection block flexibility, or $F_c$. The number of wires to which each incoming track can connect, in a switch block, is called the switch block flexibility, or $F_s$. The length of a segment is measured by the number of CAB blocks it spans. The segmentation distribution $F_{sd}$ defines what fraction of the tracks in each channel is of each length.

This FPAA architecture contains a 4X4 CAB array. Each CAB has 8 pins, with 4 input pins on the left of the CAB and 4 output pins on the right of the CAB, for fully differential circuit operation. Each CAB is surrounded by 4 connection boxes. There are 8 tracks per horizontal and 8 tracks per vertical channel. In all, there are 13 switch boxes and 32 I/O PADs, with 8 pads on each row/column of CABs. The left column and bottom row PADs are for input only; the top row and right column PADs are for output only. All the routing resources are uniformly distributed. So, for this architecture, W is 8 for all channels, $F_c$ is 8, $F_s$ is 4, $F_{sd}$ is 1 and all segments have length 4. This FPAA architecture does not contain segmentation but it can easily be modified, if segmentation is desired. Any array based FPAA can be readily fitted into this basic architecture, with some appropriate adjustment. A more versatile structure can be obtained by adding more pins, pads, tracks or segmentation.

The coordinate system for the architecture is, as defined in Figure 2, from (0, 0) to (5, 5). The four corner positions, (0, 0), (0, 5), (5, 0), (5, 5), are blank areas, i.e. no routing resources are available. Each X or Y directed channel belongs to the pad or CAB right below it, or on the left to it, having the same coordinates. In the routing resource graph, a pin-pad-track (PPT) number is used to record the internal index of CAB pins, I/O pads and tracks in the channel. The PPT number for PADs ranges from 0 to 7, starting with bottom or left most PAD. CAB pins are sorted from inputs to outputs. Input CAB pins have PPT numbers ranging from 0 to 3; output CAB pins have PPT numbers ranging from 4 to 7. The top left pin (first CAB pin) has a PPT number of 0, and the bottom right (the last CAB pin) has a PPT number of 7. Inside each channel, the PPT number ranges from 0 to 7, with 0 always denoting the bottom or left most track.

### 3.1.2 Switch Box and Connection Box

As depicted in figure 3.3 (a), the input/output pins of the CAB connect to the tracks in horizontal and vertical channels through a connection box. Connection boxes are also used to connect I/O PADs to the tracks. Connections from vertical to horizontal tracks, or vice versa, are switched at the intersection by a switch box.

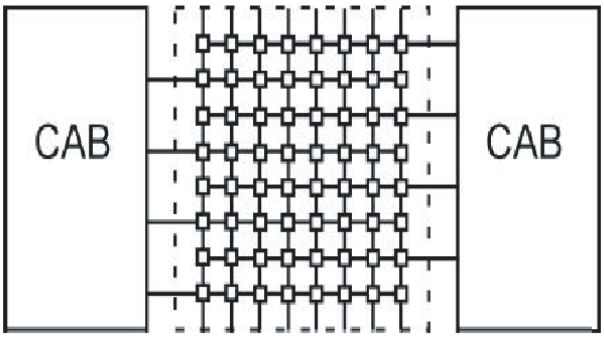
There are two types of switch box patterns. Pattern 1: tracks with different parity indices are connected. Pattern 2: tracks with same parity indices are connected. In the FPAA architecture, these two patterns alternate in each column,

starting with switch box pattern 1, which is the first one on top left.

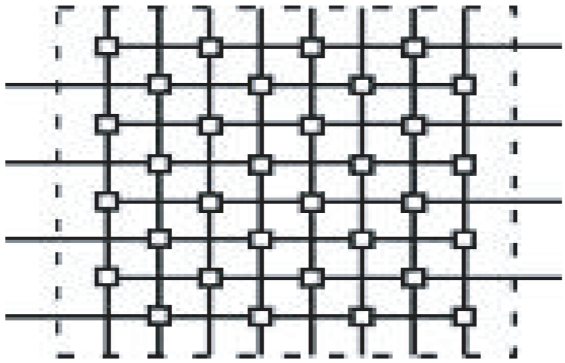
### 3.1.3 Definition of Legal Connections

Based on the architecture above, the following are defined as legal routing connections:

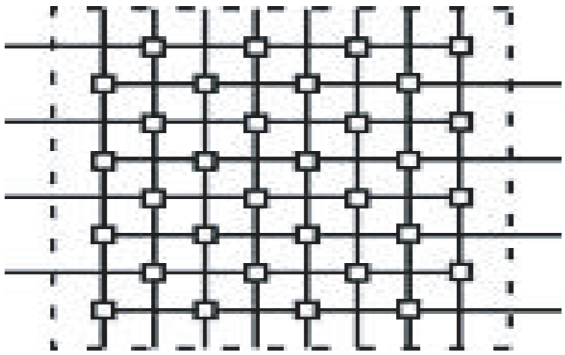
- LHS column and bottom row pads are for input only.
- RHS column and top row pads are for output only.
- LHS pads can connect to all the tracks in chany (0, 1).
- RHS pads can connect to all the tracks in chany (4, 1).
- Bottom row pads can connect to all the tracks in chanx (1, 0).
- Top row pads can connect to all the tracks in chanx (1, 4).
- Pins on the left the CAB are for input; pins on the right of the CAB are for output.
- Input CAB pins can connect to tracks in the channels immediately on the left, top and bottom of the CAB.
- Output CAB pins can connect to tracks in the channels immediately on the right, top and bottom of the CAB.
- Tracks in the horizontal channel can connect to tracks in vertical channel if a switch is available at the intersection.




(a)



(b)



(c)

Figure 3.3: (a)A Connection Box; (b)Switch box patterns 1; (c) pattern 2

- Direct connections between CAB pins are not allowed.
- Direct connections between PADs and CAB pins are not allowed.
- Dogleg is not allowed, i.e., CAB pin cannot be acted as intermediate vertex to route a net.

## 3.2 Problem Formation

Routing problems are generally studied as a graph problem [39]. All routing resources and their relationships, capacities and constraints are incorporated into a routing resource graph (RRG). The router uses this graph to solve the routing problem. A simplified FPAA architecture, and its associated RRG, is shown in figure 3.4. Each track, PAD or CAB block pin is represented by a vertex in the RRG. Each switch is represented by an edge. For examples, pin3 of CAB1 block is represented by vertex (3); wire b is represented by vertex (b). The red net is shown as a red tree in the RRG. Each vertex has a capacity, which is defined as the maximum number of nets that can use this vertex in a legal routing. Track segments have capacity one because only one net can use each. Because the Laser Makelink switch is bi-direction, the RRG of FPAA is a non-directed acyclic graph.

To route a multi-terminal net in minimum distance or delay is equivalent to finding a minimum-length tree on the routing resource graph, that spans all the connecting vertices of the net[39]. This is essentially a Minimum Steiner Tree

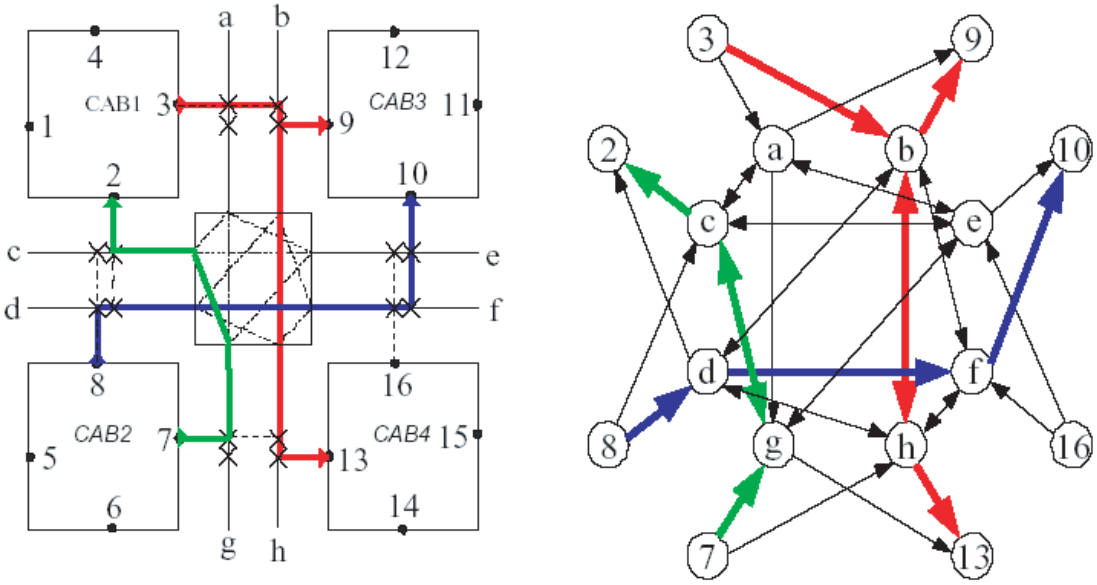



Figure 3.4: (a) a simplified FPAA architecture (b) the corresponding routing resource graph (RRG)

Problem (MST). Using RRG, the routing problem is converted into a graph problem: find multiple MSTs in the routing resource graph. The MST problem is NP-complete [41], [42], [43]. Therefore, routing multiple nets with multi-terminals, for an FPAA, is also a NP-complete problem. Accordingly, no routing algorithm can guarantee the optimal result, i.e. it's likely only an approximation/sub-optimal solution will be obtained.

There are two standard ways to store a RRG: as a set of adjacency lists, Fig. 7(b); or as an adjacency matrix, Fig. 7(c) [41]. An adjacency-list was used for the routing algorithm development, because it provides a more economic way to store sparse graphs. The adjacency-list representation of a graph $G = (V, E)$ consists of an array of v lists, one for each vertex in V. For each $u \in V$, the adjacency

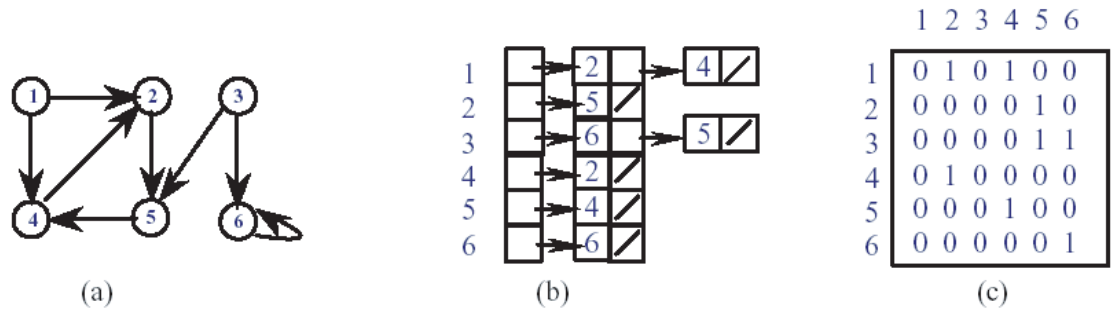



Figure 3.5: (a) a directed graph (b) adjacency list (c) adjacency matrix

list contains all the vertices, $v$, such that there is an edge $(u, v) \in E$. A potential disadvantage of the adjacency-list is that there is no quicker way to determine if a given edge is available in the graph.

A unique RRG is required for routing each FPAA architecture. Manually creating such graphs is very time-consuming, or even impossible. In order to test as many architecture variations as possible, and interactively optimize both the architecture and router, a routing resource graph generator (RRGG) was developed to automatically generate the RRG, for each given architecture. The role of the

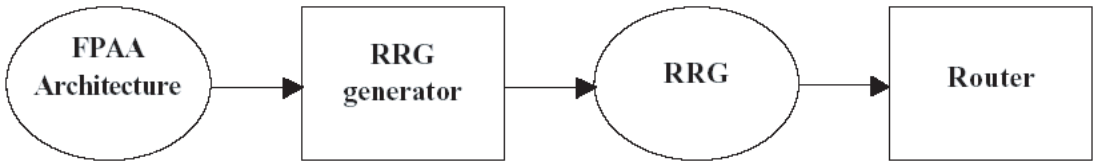



Figure 3.6: The role of routing resource graph generator

RRGG is schematically demonstrated in figure.6[40]. The RRGG converts the targeted FPAA architecture into a highly detailed RRG, which will be used by the router. The RRGG is transparent to the "user" (who defines the architecture) and

the router. Moreover, if the architecture is modified, only the RRGG needs to be modified; the router code does not need to be re-written and can still function correctly, with very little modification.

When building the RRG, a coordinate system must be clearly defined. The order or index of the vertices is chosen from bottom to top, and left to right, i.e., (0, 0), (0, 1) ...(0, 5), (1, 0) ...(1, 5), ...(5, 5). The four bland positions, (0, 0), (5, 5), (0, 5), (5, 0), should be skipped. The program starts building routing resource graph from position (0, 1). I/O PAD vertices are added onto the RRG first. Whenever there is a possible connection between PAD and routing tracks, an edge (i.e., a neighbor of this vertex) is added into the linked list of that PAD vertex. Since Makelink switch is bi-directional and this is an undirected graph, an edge is also added into the linked list of this vertex's neighbor, as well. This work is done by subroutine *creat_edge_list*. Given the vertex (x, y) coordinates, its routing resource type and its internal PPT number, vertex index in the routing resource graph can by calculated by calling subroutine *get_vertex_index*. X, Y, routing resource type and PPT number can be directly obtained from the loop control. For example, PAD 0 (the first PAD in PAD group (0, 1)) is added into the RRG first. According to our connection definition, it can connect to all the tracks in Channel Y (0, 1). The program loops over all the tracks at position (0, 1), from 0 to 7, calculates their indices respectively, and adds these vertices into the neighbor list of PAD 0. At the same time, PAD 0 is added into the neighbor list of those tracks. Similarly, CAB pins and the associated X/Y tracks are added. If there is a switch box at the intersection of the X and Y channels, the tracks in these channels are added onto each other's

linked list. Please note, there are two types of switch box pattern. Care should be taken when adding the tracks into the neighbor list of the connected tracks. Finally, after all the vertices have been counted, the generated RRG is outputted into an RRG file.

## 3.3 FPAA Routing Algorithm

### 3.3.1 Introduction

Conventionally, the task of routing is carried out in two phases: global routing and detailed routing [39], [40], [44], [45], [46], [47]. In the global routing phase, a list of regions (channels) are assigned to each net, without specifying actual track-pin connections; connections are completed in the detailed routing phase. This two-step routing method is mainly due to the complexity of the problem. However, there are two apparent drawbacks: (1) The task of detailed routing is usually very difficult or impossible because the routing resource of FPGA/FPAA is fixed and limited and the detailed routing is highly constrained by the decisions made during the global routing phase; (2) In case the circuit is routable, it's very likely the routing result is only sub-optimized, even if an optimized result in both phases were obtained. Therefore, a one-step, combined, global-detailed routing scheme is preferred in our routing algorithm development [48], [49], [50], [51].

As stated previously, the routing problem is essentially an MST problem, in graph theory. There are several algorithms available to attack this problem. Many of these routing algorithms use some variations of Lee's Maze router. A Maze router

essentially consists of running Dijkstra's algorithm. The searching strategy is very similar to the one used in Prim's algorithm. So, in this subsection, a brief overview of these three most important algorithms is given.

- Lee's Maze Algorithm[52] This algorithm is best illustrated by figure 3.7. The

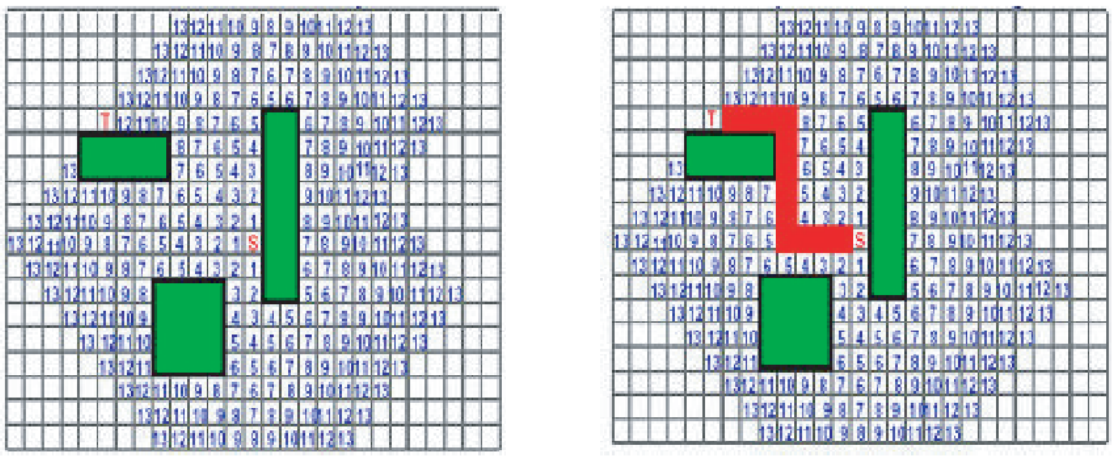


Figure 3.7: Lee's Maze Router

task is to find a shortest path from source, s, to target, t. First, grids overlaid over the plane are defined. Each grid is where one wire can cross. Then mark each grid by its relative distance to the source. The search begins at the source, finds all the grids at distance 1, distance 2 ... until reaching the destination, $t$. This algorithm addresses the problem in a manner consistent with wave propagation. With this procedure it is guaranteed that the shortest path will be found.

- Dijkstra's Algorithm[41] Dijkstra's Algorithm solves the single-source, shortest-path problem on a weighted, directed graph $G = (V, E)$, for the case in which all edge weights are non-negative values, and is presented, as follows: Dijkstra's

```
Dijkstra(G,w,s)
1. for each u∈V[G] {
2.          dist(s,u) = ∞;
3.    pre(u) = NULL;
   }
4. dist(s,s) = 0;
5. Done = Φ;
6. Q = G;
7. while Q != Φ {
8.    find u ∈ Q with min. dist(s,u);
9.    Q = Q – {u};
10.  for each v adjacent to u
11.     if dist(s,v) > dist(s,u) + dist(u,v) {
12.  dist(s,v)=dist(s,u)+dist(u,v);
13.         pre[v] = u;
     }
14.  Done = Done∪ {u};
15.}
```

Figure 3.8: Dijkstra algorithm

algorithm maintains a set "$Done$" of vertices whose final, shortest-path from the source $s$, have already been determined. Initially, all the vertices are enqueued to $Q$. The algorithm repeatedly selects the vertex $u \in Q$ - Done with the minimum shortest path evaluated, saves its predecessor if available and inserts $u$ into set "$Done$".

- Prim's Algorithm[41] Prim's algorithm operates much like Dijkstra's algorithm for finding shortest paths. At each step, a light edge is added. The shortest path of a new vertex is calculated, with respect to the existing, partially finished tree (net). This algorithm applies a greedy strategy. The key to efficiently implementing Prim's algorithm is to make it easy to select a new edge to be added to the tree. During execution of the algorithm, all vertices that are not in the partial tree (net) are stored in a priority queue. Key $v$ is

```
Prim (G,w,r)
1. for each u∈ V[G] {
2.          do key [u] <- ∞;
3.        π [u] <- NIL
4. key [u] <- 0
5. Q <- V[G];
6. Q = G;
7. while Q != Φ {
8.    do u <-  Extract Min (Q)
9.    for each v ∈ Adj [u]
10.      do if v ∈ Q and w (u, v) < key [u]
11.              then π [v] <- u
12.           key [v] <- w (u, v)
```

Figure 3.9: Prim algorithm

vertex's priority value. Prim's algorithm is shown as above.

### 3.3.2 Pathfinder Negotiated Routing Algorithm

There are many trade-offs when routing a circuit netlist. For example, performance and congestion may conflict. A pure, routability-driven router may produce poor performance, while pure performance-driven routing may result in an unroutable circuit. How to balance these trade-offs is the major concern of the router. A very efficient way to do this, is to incorporate those trade-offs into a cost function. Most routers perform multiple routing iterations in which some or all of the nets are ripped-up and rerouted by different paths to resolve competition for routing resources, or to improve circuit performance. The criteria to determine which net should be routed first, is determined by the cost function. Therefore cost function design is critical for routing algorithm development.

The FPAA router is based on the Pathfinder Negotiated Routing Algorithm

[48], [49],[51], [53]. Depending on the cost function design, it can be either pure routability-driven or balanced, congestion-performance driven routing. However, for this small scale FPAA, a 4x4 CAB array comparable to Anadigm's AN10E40, a routability driven router is sufficient.

- Cost Function Definition [48], [54]

  Before any further discussion of the algorithm, let's first define the cost function. The following equations were used for the cost function in this router:

$$Cost(n) = b(n) \cdot h(n) \cdot p(n) \tag{3.1}$$

  where $b(n)$, $h(n)$ and $p(n)$ are base cost, history congestion cost and present congestion cost, respectively. The present and history congestion cost functions are defined, as follows:

$$p(n) = 1 + occupancy \cdot p_{fac} \tag{3.2}$$

$$h(h) \quad = \quad h(n)^{i-1} + occupancy \cdot h_{fac} \tag{3.3}$$

  where $p_{fac}$ and $h_fac$ are experimental parameters, and $i$ is the iteration number. When $i = 1$, $h(n)$ equals 1. The example in figure 3.10 shows how the router can use $p(n)$ to resolve the congestion. During the first iteration, all 3 nets go through vertex $B$, with lowest cost. During subsequent iterations, $p(n)$ is updated, i.e., the penalty of using vertex B increases. Then, during some later iteration, net 1 will find that a path through vertex $A$ gives a lower cost. Similarly, net 3 will find that a path through $C$ gives overall lower cost.

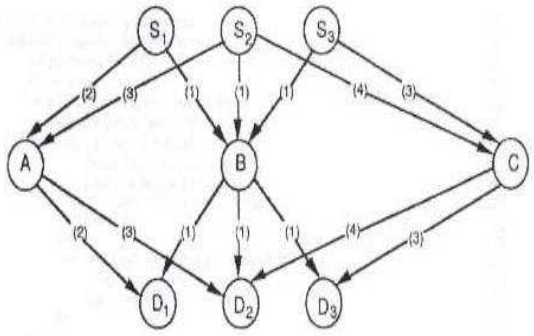



Figure 3.10: The functionality of p(n) in resolving the congestion

In this router, base cost is set to 1, for all the vertices. The performance of the router is not very sensitive to how the exact base cost is chosen, since the primary goal of the router is congestion avoidance, regardless of the base cost value. In the $p(n)$ and $h(n)$ functions, $p_{fac}$ and $h_{fac}$ are two parameters that determine how the routing is scheduled. Since h(n) is incremented after every iteration and provides sufficient penalties for overused vertices, hfac can be set to a constant value. $h_{fac}$ is set to 0.5 in this router. $p(n)$ is updated more frequently. To achieve high quality routing results, $p_{fac}$ should initially be small, allowing congestion to have little penalty; and gradually increases from iteration to iteration. The trade-off is that slowly increasing pfac will get a better quality routing, while quickly increasing $p_{fac}$ (by making congestion very expensive) will speed up the router. Here, $p_{fac}$ is initially set to 0.5 and then increase it by 1.5 times of its previous value, with each iteration. Due to the scale of this FPAA, there's no noticeable differences due to variations in these two parameters.

- Pathfinder Negotiated Routing Algorithm The detailed pathfinder negotiated

```
RT(neti): a linked list used to store the set of vertices in the current routing of net i
While (overused resources exist && max iteration not exceeded) {
    For (each net, i) {
        If RT is not empty then Rip-up existing RT(neti) and update p(n) ;
        Initialize RT to the source terminal;
        For(each sink net i) {
                If PQ is not empty then free PQ and re-initialize PQ;
                Initialize PQ to RT;
                Mark all the vertices as un-reached by wave expansion;
                Initialize PriorityQueue to RT(neti) and set pathcost equal to the base cost of
                each vertex in RT;
                If this sink j is not foundd in RT(neti) {
                    do {
                        Dequeue PQ;
                        For (all fanout vertices n of node m){
                        If (this fan-out is not a PIN or PAD and un-reached during previous
                wave expansion)
                            add it  to PQ & update pathcost(n) = pathcost(m) + cost(n);
                        else if (this fanout is a sink)
                                    add it to a sink list;
                        else continue wave expansion;
                        }
                    } while (no sink has been found); /* Wave expansion ends here */
                }
                if  ( more than one sinks are found during this wave expansion) {
                        add those sinks and their parents to RT;
                        update p(n) only if vertex n is not contained in RT;
                }
                for (all vertices in path from RT(i) to sink,j){  /* Backtrace from the linked list
        of sinks */
                        Update p(n) only if  vertex n is not contained in RT;
                        Add n to RT(i);
                } /* Backtracing ends here */
            }
    }
    Update h(n) for all n;
}  /*End of one iteration*/
```

Figure 3.11: The improved pathfinder negotiated routing algorithm

routing algorithm is shown as of above.

Pathfinder negotiated routing repeatedly rips-up and re-routers every net in the circuit until all the congestions are eliminated. During the first routing iteration, every net is routed for minimum cost, even if this leads to congestion. After each routing iteration, the cost of overuse is increased. The router can determine how to arrange the routing resource, based on the cost of each

vertex. Consequently, if overuse exists at the end of a routing iteration, more iterations are performed to resolve this congestion.

- Implementation of Pathfinder Negotiated Routing Algorithm

The router takes the netlist as its input and starts routing based on the RRG generated by RRGG. The flow chart for the program is shown in Fig.12. There

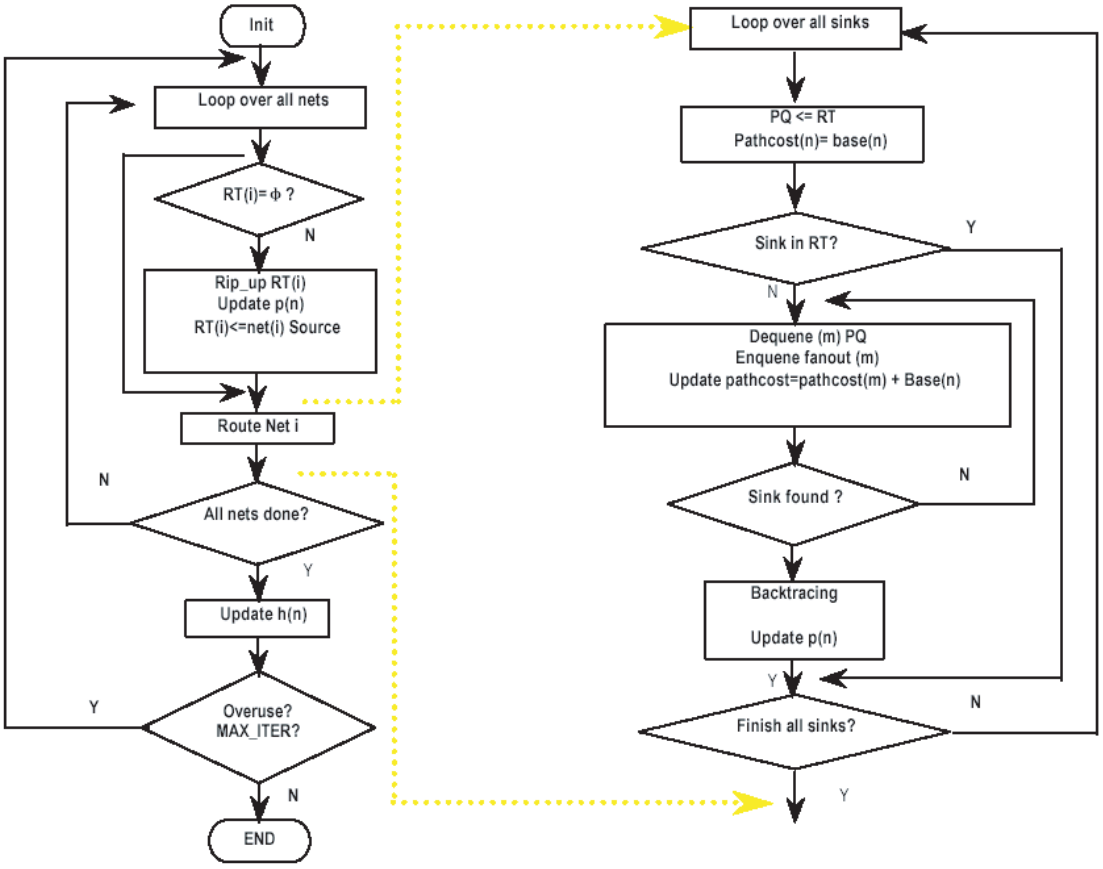



Figure 3.12: Pathfinder algorithm

are three types of vertices in the router: routing resource graph (RRG) vertex; routing tree $RT$ vertex; and priority queue $PQ$ vertex. A routing tree is used to store the vertices in the partially finished, or finally completed, routing of

a net. Each $RT$ corresponds to a net in the netlist. It will be ripped-up, after every routing iteration, until all the nets are successfully routed. In this pathfinder negotiated routing, whenever a new iteration starts, the router first rips-up the existing RT/net. The cost of vertices in the $RT$ is re-calculated and the source of this net is re-assigned to $RT$. Then, it loops over all the terminals of this net. The router performs a breadth-first search (wave expansion) over all the fanouts of the lowest cost vertex in $PQ$. If no sink is found, all of the fanouts are added to $PQ$. After the router finds a sink (or more sinks), it begins back-tracing. If $x(x > 1)$ sinks are found, the first $x - 1$ sinks and their parents are added to the $RT$, then the router starts back-tracing from the last one. Routing iterations stop when all the nets are successfully routed or when the maximum routing iteration is exceeded.

When programming the router, the following should be noticed:

- Since some vertices in $RT$ may appear more than once, when initializing $PQ$ to $RT$, it need to make sure there's no repeated vertex in $PQ$. Otherwise, multiple wave expansions will be carried out from the same vertex. Obviously, this will reduce the router's performance. Similarly, during intermediate wave expansion stages, any vertices that have been previously reached should be removed from future wave expansions.
- By intuition, if a netlist is placed appropriately in a FPAA/FPGA, sinks of the same net tend to stay close to each other. It is very likely that more than one sink could be found during the same wave expansion.

However, in the original algorithm, the wave expansion procedure stops whenever a sink is found. Then, another new wave expansion starts for the next sink. Thus a significant amount of the router's work could be wasted, especially when the wave expansion starts very deeply inside the RRG. Thus, a more efficient mechanism was developed, by introducing a temporary sink list. Every wave expansion must be fully completed even if a sink has been found. If more than one sink is found, those sinks are added to the temporary sink list and then added to $RT$. Before a new iteration starts for the next sink, the router first checks if this sink has already been contained in $RT$. The wave expansion stops when the number of sinks found is larger or equal to 1. The back-tracing stage starts from the last sink in this temporary sink list. Since sinks sometimes may be found out of the loop order, a flag variable should be introduced to ensure every sink of a certain net is found. The router checks this flag before it moves onto the next sink in the loop, so it won't miss any sink.

– Priority Queue, $PQ$, is the critical data structure in implementing this algorithm. The memory occupied by $PQ$ must be appropriately allocated and released after each iteration. In order to better manage the dynamically allocated $PQ$ memory, three special data members, $size0$, $avail0$ and $d0$, are used to track the vertices that were historically in $PQ$. Those three members represent a redundant array. This redundant array is used to copy the locations of all the vertices that are currently in $PQ$,

or that used to be in $PQ$. Then, the router knows where and how to release the memory for the new $PQ$.

## 3.4 Data Structure

The primary data structures used in the router are linked list and priority queue.

There are three types of vertices in the router: routing resource graph (RRG) vertex; routing tree $RT$ vertex; and priority queue $PQ$ vertex. RRG vertices and $RT$ vertices are maintained by linked list, while $PQ$ vertices are maintained by priority queue. Their definitions are shown as follows:

When a vertex is used by a net, its occupancy increases by 1. Capacity is 1 for all vertices, since only one net can legally use a vertex. A vertex's edge list is designed as a 1-D array, for easy access. After the main program calls the $build\_rrg()$ subroutine, all the RRG vertices will be loaded into memory and ready to use for the router.

A routing tree is used to store the vertices in the current routing of a net. Each $RT$ corresponds to a net in the netlist. It will be ripped-up after each routing iteration, until all the nets are successfully routed. Since all the information needed in the back-tracing stage is stored in priority queue, $RT$ was implemented just with a simple linked list.

The critical data structure in the routing algorithm development is priority queue, or more precisely, a minimum binary heap priority queue, [39],[41], [55]. For

```
typedef struct {int index; short x; short y;short ppt_num; t_rr_type type; int occupancy;
                int capacity; int num_edges; int *edge_list; } t_rr_vertex;
 / * index: index of the vertex                                    *
 * x, y:  integer coordinates                                *
 * type:  What is this routing resource?                     *
 * occupancy: how many nets are using this vertex now?       *
 * capacity: how many nets can legally use this vertex?      *
 * ppt_num:  Pin, track or pad number, depending on rr_vertex type.   *
 * num_edges:  number of edges exiting this vertex, i.e. the number    *
 *           of vertexs to which it connects.                          *
 * edge_list: pointer to the linked list of all its neighbors        *
*************************************************************************/

struct s_RTvertex { int index; short PQflag; struct s_RTvertex *pNext; };
typedef struct s_RTvertex t_RTvertex;
/**** Data structure of a routing tree member *****/
/* index: the index of this vertex; */
/* pNext: pointer to the next vertex */
/*******************************************************/

struct s_PQvertex {    int index;       struct s_PQvertex *pParent; double pathcost; };
/**** Data structure for priority queue vertices *****/
/* index: the index of this vertex; */
/* pathcost: the pathcost of this vertex in the partial net */
/* pParent: parent of this vertex in RRG, NOT the parent in PQ*/
/*******************************************************/
```

Figure 3.13: Data structure definitions

a regular queue, new items are added to one end of the queue and are removed from its other end. The sequence an item is taken out of the queue is first-in-first-out (FIFO). A priority queue is different from a regular queue in that the items it contains are not arranged in the order of their respective time of enqueuement, but by their priority. When an item is removed from a priority queue, it has, of all items, the highest priority (in the context of this router, the highest priority means the item has the minimum pathcost).

A binary heap basically is a binary tree for which the following two properties hold:

- Each vertex is associated with a scalar key value, called priority.
- No vertex in the three has children whose key is higher than its own.

Binary heaps have two important properties. First, the vertex bearing the highest key value is always the root vertex. Second, insertion or removal of records takes $O(\log n)$ time, where $n$ is the number of items in the heap. A binary heap priority queue is a priority queue, internally using a binary heap to organize its items.

There are many methods to implement a priority queue. The most efficient way is to use a plain, $1 - D$ array. Assume there are n vertices in $P$. The vertices are stored in the array, with $n$ slots in which:

- the children of the vertex in slot $i$ occupy slots $2i$ and $2i + 1$
- the parent of the vertex in slot $i$ lives in slot $i/2$.

So, when removing the lowest cost vertex from $PQ$, the root vertex that sits in slot 1 is going to be removed. There is a straightforward one-to-one correspondence between binary heaps and flattened-out array representations of binary heaps. Since the link relationship between any two vertices is directly obvious from their respective slot indices, it's no need to explicitly store any links, thus saving substantial amounts of time and space.

## 3.5 Investigations of Performance Constraints on the Routing

The goal of routing is not only to complete all the required connections without congestion, but also to satisfy a set of performance constraints. For a small

scale FPAA (comparable to Anadigm's AN10E40 [4]), a routability-driven router is sufficient. When the scale of FPAA grows and bandwidth of instantiated circuit increases significantly, performance-driven routing would be necessary.

The performance constraints imposed on analog routing are quite different from that of digital routing. For an FPGA/digital circuit, performance is measured by clock speed or/and delay on the critical path. However, for an FPAA/analog circuit, the system performance is usually measured by its bandwidth, gain, slew rate, output swing, CMRR, PSRR, linearity etc. Thus, signal delay is not the only concern. Routing parasitics can affect the performance of analog system in many different ways. For examples: (1) In an op amp circuit, a small capacitive coupling may degrade the frequency response due to the Miller effect; (2) Stray coupling which gives rise to positive feedback may lead to oscillations. (3) In some cascode configuration, the output node usually has very large resistance, Rout. When a net travels a long distance, the parasitic capacitance to ground can introduce an extra pole (for example, a pole very close to the dominant pole) that may deteriorate the op amp's stability and slew rate.

To our best knowledge, there is no explicit timing definition comparable to the digital counterpart. Therefore, FPAA routers cannot simply compare the timing criticalness/delay of two paths to decide the route. In the digital domain, the performance constraints are in fact induced by RC delay, which can be counted efficiently with the timing term in the cost function. However, the performance constraints (tolerable variation of gain, bandwidth etc.) imposed on analog array are too abstract for the routing tools to handle directly; thus they must be converted to

a set of routing constraints, i.e. interconnect parasitic constraints. Once the routing constraints are met by the router, the performance constraints of the analog circuit should also be satisfied. The performance-driven routing problem can be defined as follows [56], [57]:

***Definition***: *For a set of performance functions* $\{Wi\}$, $i = 1, 2, \ldots N_w$ *and a set of parasitics* $\{p_j\}, j = 1, 2, \ldots, N_p$, *The parasitic constraints or routing constrains on a subset of* $\{p_j\}$ *are defined as*:

- *Matching constraint*: $p_j = p_k$
- *Bounding Constraint*: $p_j \leq p_{j\_bound}$

*and they ensure:* $\Delta W_i \leq |\Delta W_{i,max}|$, *where* $|\Delta W_{i,max}|$ *is the maximally allowed performance variation due to the parasitics.*

Parasitics that are to be controlled during routing are metal wire resistance, switch resistance, metal wire to ground and metal-to-metal capacitance.

Modeling the interconnect as a true, frequency-dependent transmission line can capture the behavior of the line more accurately. However, inductance is much more complicated to extract than resistance or capacitance because of the loop current definition of inductance. The critical length of a line can be determined from knowing the desired signal frequency along with the speed of propagation of the interconnect structure. As a rule of thumb, an interconnect structure should be considered as a transmission line when its physical length approaches 1/4 to 1/10 the wavelength of the highest frequency signal [58], [59].

For the case of a simple microstrip line, the wavelength at a given frequency

is:

$$\lambda_g = \frac{300}{F\sqrt{\varepsilon_{eff}}} mm \tag{3.4}$$

where $\varepsilon_{eff}$ is the effective dielectric constant given by $\varepsilon_{eff} = 1/2(\varepsilon_r + 1)$, and $F$ is the frequency in GHz. Assuming the highest frequency signal of interest (to pass) is 1 GHz, the corresponding wavelength is 191.69 $mm$. Assuming the 1/4 rule, the line length for which transmission line property becomes important is about 47.92mm. Therefore, the effect of parasitic inductance can be neglected for on-chip circuits.

Due to the unique advantages of laser Makelink technology, parasitic capacitance of the interconnect metal wires (figure 3.14) is the major concern in FPAA routing. Parasitic resistance can also be taken into account, if needed.

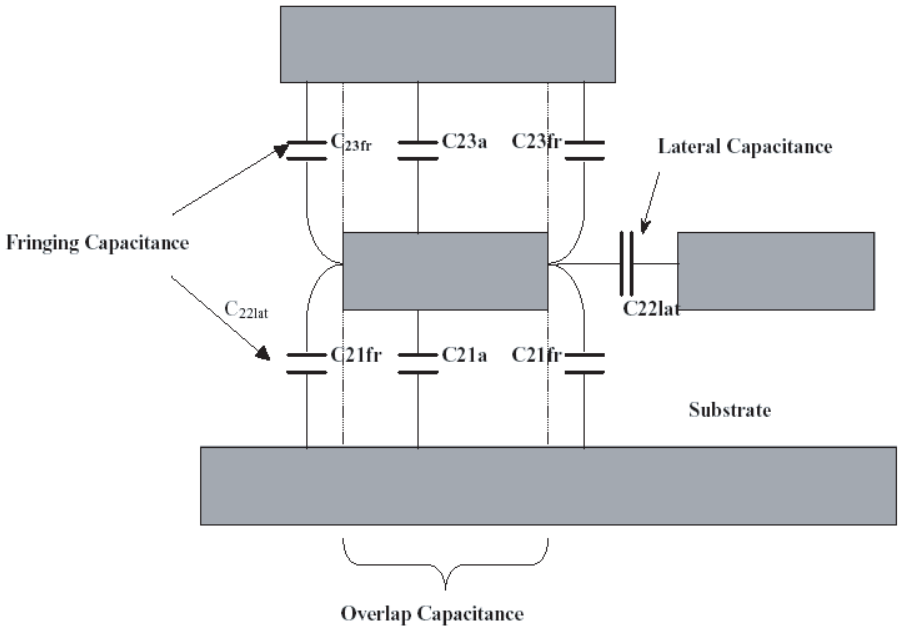



Figure 3.14: Pathfinder algorithm

**Imposing Bounding Constraints on Performance-driven Routing**

Bounding constraints can be divided into two classes: (1) loading constraint (to ground); (2) coupling constraint.

(1) Loading constraint: Usually the routing interconnects reside on the top level metal layers. In many cases, the parasitic capacitance to ground $C_{ground}$ is not a problem. But, if the metal wire travels a long distance (the net spans a large portion the FPAA chip), $C_{ground}$ can deteriorate op amp performance, such as stability and transient response time, especially when the circuit node has large impedance. So, when routing a net, its accumulated parasitics are checked against the pre-defined bound. If the bound is exceeded, the wave expansion terminates and starts over again.

(2) Coupling constraints: for analog circuit, the coupling capacitance could be more important than the parasitic capacitance to ground, since it usually has a much larger value. The sub-problem can be defined as: given a set of sensitive pairs of nets $(n_i, , n_j)$ (sensitive pairs are pairs of nets between which coupling constraints are imposed) and a set of associated bounds $C_{bound}(i, j)$, the completed routing should satisfy: $C(i, j) \leq C_{bound}(i, j)$, where $C(i, j)$ are coupling capacitance between nets $n_i$ and $n_j$.

Capacitive coupling is present whenever two nets have segments that cross or are parallel to each other. Thus, it can be further classified by crossover constraints and adjacency constraints. For FPAA, the adjacency constraints are the dominant factor because most of the capacitances induced by crossover can only occur at the intersections of horizontal and vertical channels. A preliminary idea of imposing the coupling constraints on the routing, is: when the routing of one net in the sensitive

pair is completed, the cost of those tracks that cross over it or are immediately close (some influence distance should be set; to the first order, only consider the closest ones) and parallel to it increases. The increased value must be larger than the regular cost due to congestion. This will make the router tend to use other tracks, which have no or little coupling capacitance, to route another net in the sensitive pair. Then, net re-ordering is performed after each iteration. With this approach, the effects of coupling constraints are effectively incorporated into the cost function.

**Imposing Matching Constraints on Performance-driven Routing**

Fully differential topology is frequently used in the FPAA circuit. This results in an additional need for the interconnect parasitics associated with appropriate nodes or branches to nominally match, for impedance matching and noise cancellation purposes. Bad matching not only reduces the CMRR but also increases the offset voltage, or even affects proper functioning of the circuit. The matching constraints require: (1) For impedance matching, the capacitances to ground associated with each matched pair of nets should be equal; (2) When a casual net (the net that does not have any constraints) is close to a matched pair, the coupling capacitances between that casual net and the pair of matched nets should match; (3) When two pairs of matched nets come close to each other, it is necessary to match the direct-coupling capacitances and cross-coupling capacitances. Besides having symmetrical loading, this also ensures that equal levels of noise on the two nodes of one matched pair causes the same on the other pair, if any coupling is present. The FPAA router can employ a simple scheme to route the matched pairs. First, net ordering is performed. Then the pair of nets in each matched pair is treated as a single net (called

merged net) and routed. Another way to impose matching constraints is, after routing one net of the matched pair, the cost of those tracks that are symmetric to the segments of the finished net can intestinally decreased. The router will tend to use these "matched tracks" to finish the routing, so that matching constraints are also effectively incorporated into the cost function.

For the current FPAA architecture, the routability-driven router is sufficient because:

(1) The scale of the FPAA is quite small. The specifications of the CAB and the targeted application speed is still well below 100 MHz range.

(2) The pathfinder algorithm employs a similar strategy as Lee's Maze algorithm, which is used to solve the shortest path problem. Thus, although the router developed is "congestion-only driven", it in fact not only resolves the congestion but also tries to find the "shortest path". In other words, the accumulated parasitics (especially the loading capacitance and serial resistance) are automatically kept to a near minimum value, along with the wave expansion process.

Appendix B is a brief program documentation for the FPAA router. The output is the laser Makelink switch indices, which can be converted into physical $(x, y)$ coordinates on the actual layout.

# Chapter 4

# Configurable Analog Block

The configurable analog block, i.e., CAB, is a critical architecture building block. The FPAA developed in this work is essentially an array of CABs. They are connected by the surrounding interconnect network, including horizontal channels, vertical channels, connection boxes and switches boxes, to route signals between CABs and I/O pads. The CAB circuit and its internal arrangement strongly affect the flexibility and functionality of the FPAA. It is always desirable to implement an application just using one CAB or as few number of CAB's as possible. An efficient CAB architecture can minimize the unnecessary external long routings wires, which contribute significant more parasitics than the CAB internal wiring.

An CAB is usually composed of several programmable capacitor arrays (PCAs), programmable resistor arrays (PRAs) and an op-amp-like analog core unit. In the following section, PCA and PRA topologies are discussed first.

## 4.1 PCA and PRA

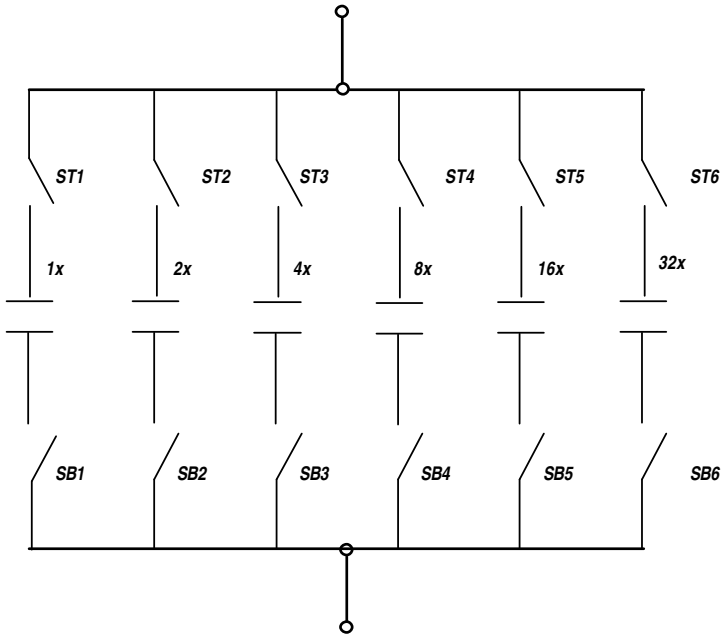
Generally, the most "expensive" parts in VLSI technology are not active devices but capacitors and resistors, because these passive components occupy a large portion of the chip area resulting a significant silicon real estate cost, and it's diffi-

cult to precisely control their absolute values. However, for continuous-time mode operation, resistors and capacitors can't be completely removed or substituted with active devices. They are used to realize feedback loop, signal coupling, integration, differentiation and other analog signal processing functions. Thus, they are a must for the proposed FPAA.

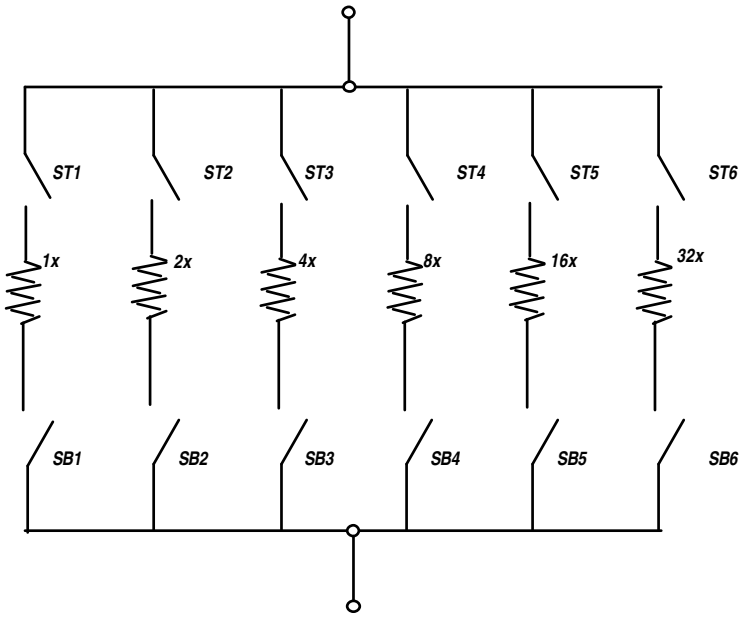
Those passive component values are obtained through the programmable capacitor array (PCAs) and the programmable resistor arrays (PRAs). To minimize area cost and increase the design flexibility, their values and arrangement inside the CAB must be chosen with special caution. In [60], the resistors in the PRA and the capacitors in the PCA and PRA are all in parallel. For each PCA or PRA, there are only two terminals. This is shown in Figure 4.1.

The drawback of this arrangement is even if only one resistor/capacitor is used, the rest of them will not be usable anymore because they share the terminals. This wastage considerably increases the chip cost because more PCAs and PRAs will be needed to increase the flexibility. Also, the way the PRA is constructed makes it impossible to obtain resistor value higher than 32x the unit resistance.

To remedy the above difficulties, a new PCA and PRA topology was developed, as shown in figure 4.2. Considering the way that the capacitance is added up, the binary-weighted capacitors in PCAs are placed in parallel. The smallest capacitance unit is denoted by 1x. The capacitors can be used individually, or users can pick any number of them or all of them to obtain larger desired capacitance by appropriately programming the switches. Then the available capacitance range achievable via PCA is from 1x to 63x with minimum resolution of 1x. By the same token, the

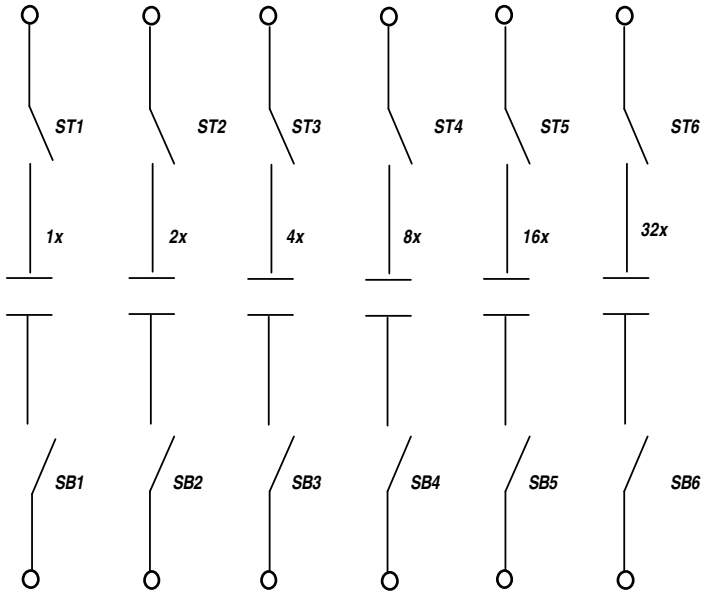



(a)

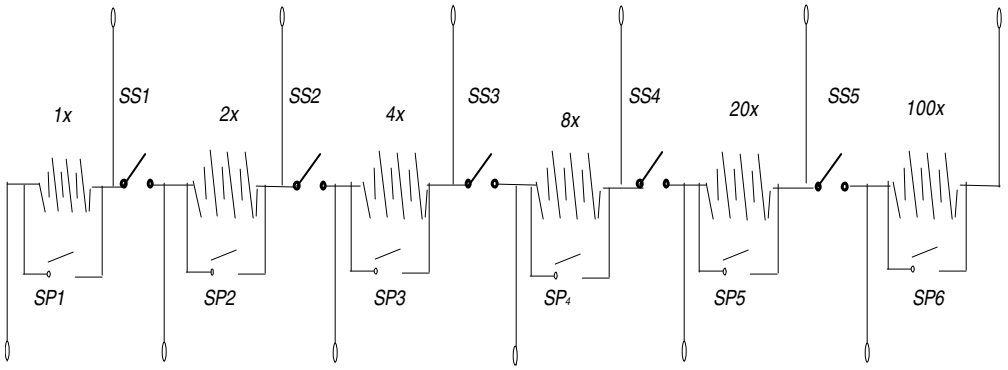



(b)

Figure 4.1: The resistors and capacitors arrangement inside the CAB [60] (a) PCA ; (b) PRA




(a)




(b)

Figure 4.2: The improved resistors and capacitors arrangement inside the CAB (a) PCA ; (b)PRA

resistors in PRA are in serial. The switches $SS_x$ and $SP_x$ (x=1,2,3,4,5,6) allow almost arbitrary connections between the resistors. For examples, if switches $SP_1$ through $SP_6$ are all closed, this PRA essentially behaves as a metal wire (0Ω) and can be used to configure a unity gain buffer. If maximum resistance is desired, one can simply close switches $SS_1$ to $SS_6$ and leave switches $SP_1$ to $SP_6$ open. If resistance of 10x is needed, switches other than $SS_2$, $SP_3$ and $SS_4$ can be left open. Also, each of the resistors or capacitors has its own terminals. Comparing to figure 4.1, this allows re-use of the PRA and PCA.

No FPAA can satisfy all application requirements. The exact unit capacitance or resistance value should be determined by the specific range of operating frequency. The basic rule of thumb is, this value should be large enough so that the parasitic capacitance of the transistors or interconnect wiring is negligible; at the mean time, it shouldn't be too large to over-load the core circuit or degrade the speed. From IC layout design perspective, all the resistors or capacitors should be built using the unit cell (if the desired resistance range is too wide, the unit cell can be made of two 2x unit value resistors in parallel).

## 4.2 CAB Structure

In this work, a fully differential difference amplifier was used as the core circuit block in the CAB(figure 4.3).

The number of PCA's and PRA's and their relative placement to the DDA should be considered for certain target applications. Considering the general use of

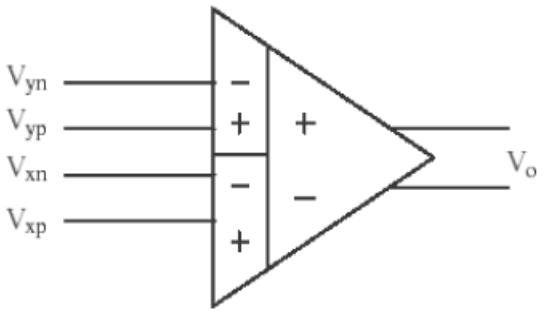



Figure 4.3: The differential difference amplifier

those resistors and capacitors, such as to form feedback loop or coupling components, two pairs of resistors/capacitors would be needed. To form different type of filter response, the capacitors should have the flexibility to be connected before the resistor or after the resistor on the signal path. As mentioned at early in this chapter, whenever it is possible, a certain application should be implemented within the CAB because of the shorter signal traveling distance thus faster speed. Bearing this in mind, an Sallen-Key bandpass filter which requires fairly complex internal wiring was chosen as a start point.

Four PCAs and four PRAs were chosen. Two pairs of them are put on the top and bottom of the DDA, which can be used to form feedback loops. Another two pairs are put before the DDA inputs. These resistors and capacitors can be uses in coupling path or in some active filter applications (Chapter 7). The overall CAB architecture is shown in figure 4.5 [62].

The regular thin black lines are single wires. The thick red lines in figure 4.5 are "BUSes". Each red "Bus" contains 6 single wires corresponding to the 6 pairs of terminals in PCAs and PRAs. Each small square in the figure represents

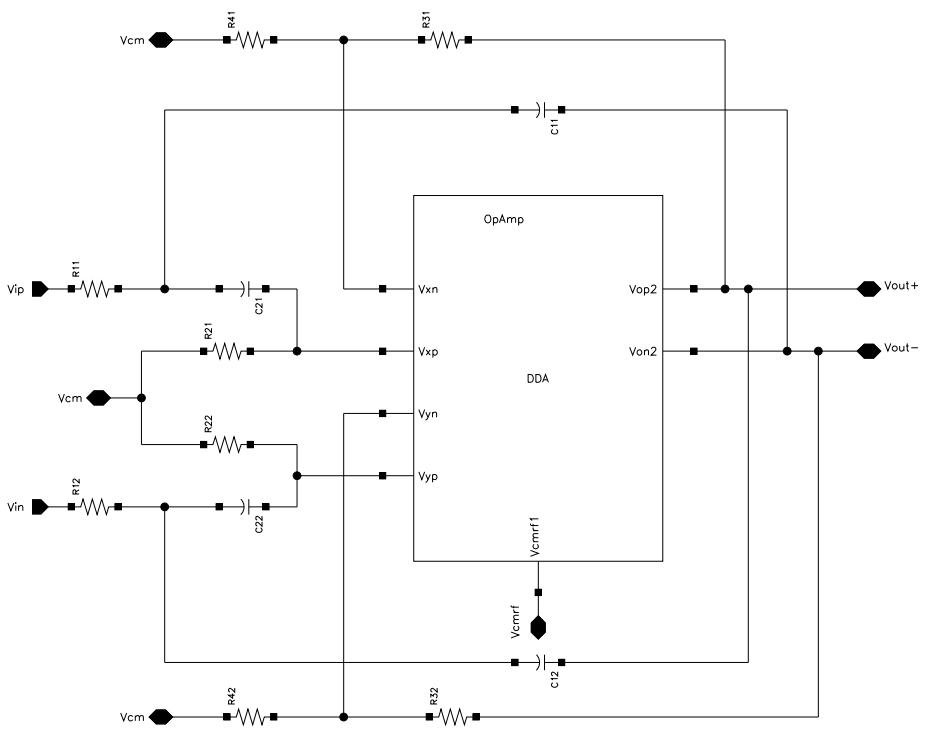



Figure 4.4: The Sallen-Key bandpass filter [61]

a programming switch or a matrix of programming switches, which is determined by the context. With this configuration, the sequence of the resistors or capacitors appearing on the signal path can be easily adjusted by properly programming the laser Makelink switch. Even one CAB is powerful enough to implement certain complex analog functions, for instance, Sallen-Key low pass filter, bandpass filter, subtracter etc, as will be introduced in Chapter 7.

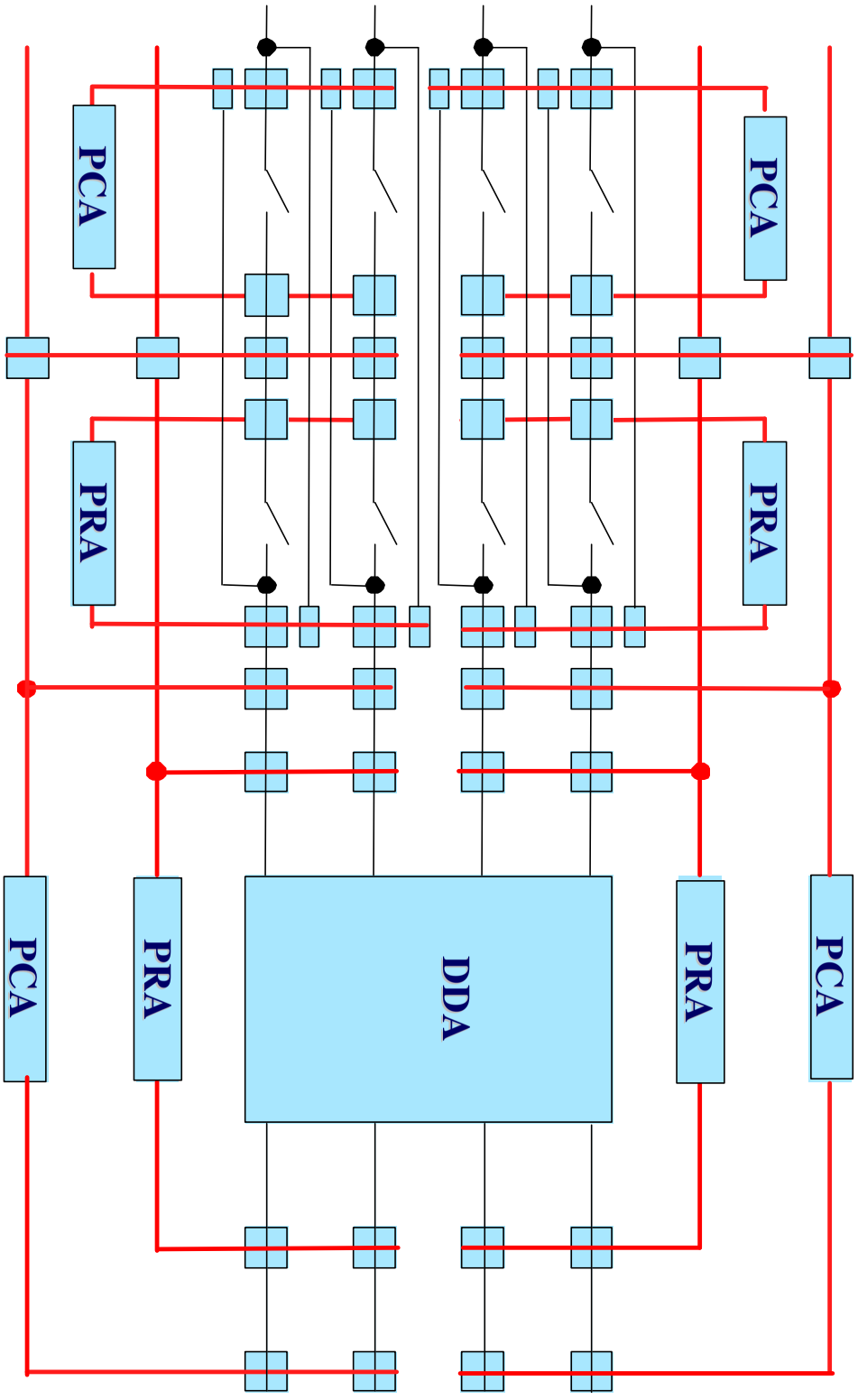



Figure 4.5: The complete CAB structure

# Chapter 5

# The Differential Difference Op Amp Design

Today's high density FPGAs usually feature a large number of modules and interconnections that allow almost arbitrary configurations of combinatorial and sequential logic. However, due to the nature of analog system design, FPAAs typically contain a relatively small number of CABs. The functionality that an FPAA can implement is largely determined by the CAB circuit. Thus a good CAB internal circuit topology not only provides more flexibility but also dramatically affects the performance of the instantiated system.

A major choice when designing an FPAA is whether to operate it in discrete-time or continuous-time. Discrete-time approaches are well suited for digital control, and for low to medium resolution, they do not require on-chip tuning scheme for VLSI implementations of the programmable components. Many discrete-time design techniques are widely used, such as switched-capacitor circuit [63], [64], controlled duty-cycle signal chopping and reconstruction [65], analog to digital conversion followed by digital processing and digital to analog conversion [66], or switched-current circuits [67]. However, such sampled-data techniques require that input signals be band-limited to at least one half of the sampling frequency (Nyquist Theorem [68]), and hence anti-aliasing and reconstruction filters are needed. This requirement

significantly limits the bandwidth of discrete-time FPAA circuit implementations. Continuous-time circuit techniques [69], [70], [71], [72], [73] do not require band-limited input signals, but may need more complicated implementations to have circuit components programmable over a large dynamic range. Continuous-time techniques of both sub-threshold [74] and linear circuits have been used in programmable analog circuits. The sub-threshold approach, however, is difficult to apply to a wide variety of analog circuits because of its increased sensitivity to process variation and the parasitic effects.

Table 5.1: Comparison between continuous time and discrete time

| **Continuous time** | **Discrete time** |
|---|---|
| No pre and post filtering | Pre and post filtering |
| No sample and hold | Sample and hold |
| Limited by op amp's bandwidth | Limited to less than $1/10^{th}$ the op-amp's bandwidth |
| Narrower component parameter range | Wider component parameter range |
| No clock noise | Noise due to clock signals |
| Less routing | Programmable routing for clock |
| Sensitive to switch nonidealities | Not sensitive to switches |

As discussed in the previous chapter, an CAB usually contains some passive component arrays (i.e., PCA's and PRA's), some interconnect switches, and an op-amp-like unit. This unit is the core circuit building block of FPAA. Its functionality and performance will dramatically affect the CAB and the overall system specifications.

## 5.1 Op Amp Topology Selection

Just like any other analog circuits, the design of an op amp is a multidimensional problem that involves many trade-offs (figure 5.1). The choice of the topology is highly dependent on the desired specifications. No op amp is suitable for all application needs, because sometimes different specifications may impose conflicting requirements on the design. For examples, gain usually trades for bandwidth; speed usually trades for power. The op amp developed here is used as the core building block for a general purpose FPAA, not for one specific application. Therefore some typical op amp parameters were optimized, while some others were not. Because there's no well-defined application standard, instead of giving a set of rigorous numbers, the following specifications of interest were proposed:

- Flexible Functionality: the op amp should be easily configurable to implement many analog functions.
- High Gain: for better linearity and precision. The desired gain should be larger than $80dB$
- High Speed: desired unity-gain frequency $f_u \geq 100MHz$ with high slew rate $SR \geq 100V/uS$.
- High Swing: for large dynamic range and high signal to noise ratio.

Other important specifications include fast settling, high common-mode-rejection-ratio (CMRR) and power-supply-rejection-ratio (PSRR) large output swing (close to rail-to-rail), good stability (phase margin $PM \geq 45/^{\circ}$) etc. The op amp is not

meant to operate at ultra low power, low supply condition, so power consumption and low noise are not the major concerns for the prototype FPAA. Again, the design were carried on the TSMC 018 CM mixed-signal CMOS process.

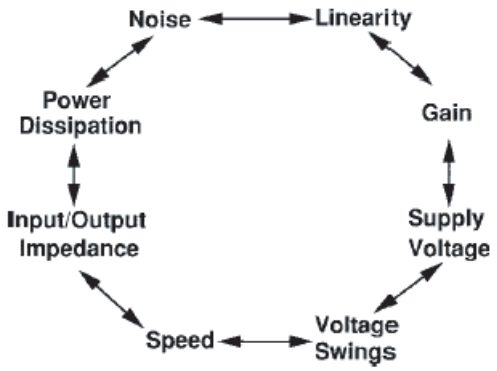



Figure 5.1: Analog Design Tradeoffs

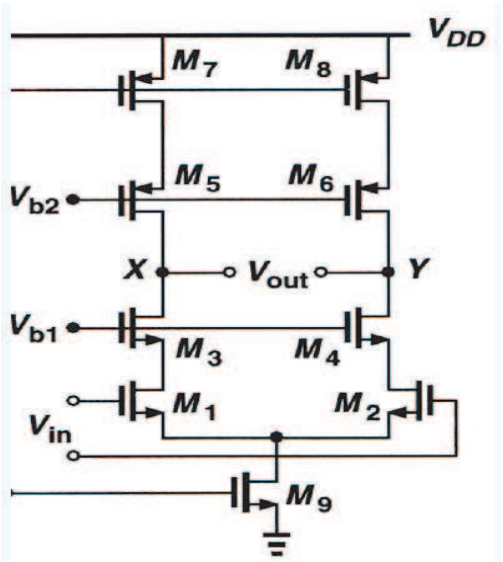
In the FPAA, all the interconnect wirings are pre-defined and fixed, the coupling effect and noise can be a serious issue. Naturely, when designing the op amp, a fully differential configuration is desired because (1) it has large output swing; (2) circuit is less susceptible to common-mode/coupling noise; and (3) there are no even-order harmonics thus better linearity [75].

There might be many ways to start the design to meet the above specifications. Probably it's easiest to start from the gain requirement. As CMOS technology migrates to deep submicron regime, the op amp design becomes increasingly challenging as the supply voltage and transistor channel lengths scale down with every generation, but threshold voltage does not accordingly.

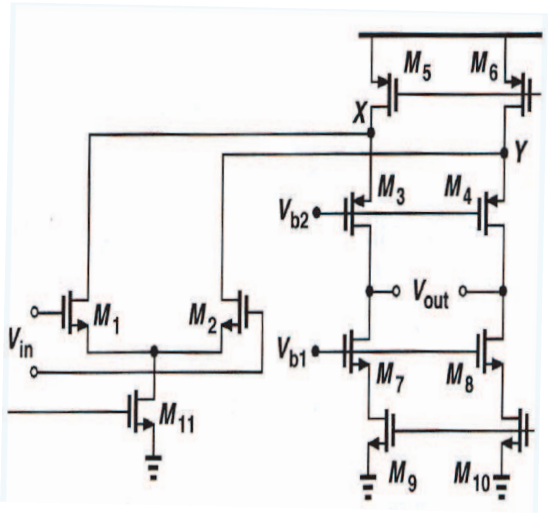
The intrinsic gain of an MOS transistor can be expressed as:

$$A_i = g_m \cdot r_o = \frac{2L_{eff}}{V_{gs} - V_t} \cdot (\frac{\partial x_d}{\partial V_{ds}})^{-1} = \frac{2}{V_{ov}} \cdot \frac{1}{\lambda} \tag{5.1}$$

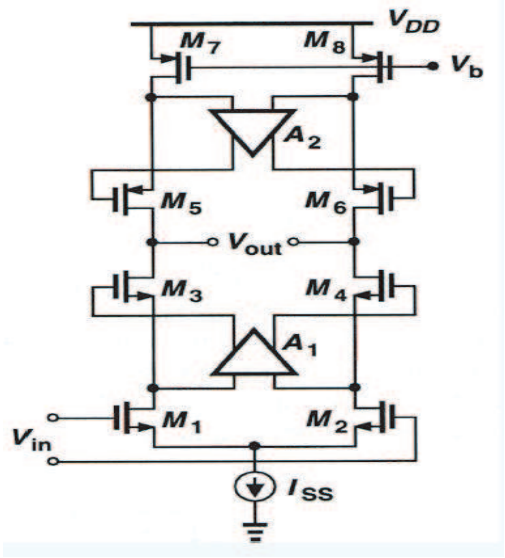
where $g_m$ is the MOS transconductance, $r_o$ is the transistor output resistance, $x_d$ is the width of the depletion region between the end of the channel and the drain, $L_{eff}$ is the effective channel length, $V_{ov}$ is the overdrive voltage ($V_{gs} - V_t$) and $\lambda$ is the channel length modulation coefficient. As the device feature size decreases, the effective channel length shrinks so much that the channel length modulation effect ($\lambda$) becomes very prominent. Usually the overdrive voltage is in the order of several hundred milivolts. $\lambda$ for short channel devices could be larger than 0.2. Thus the intrinsic gain of a short channel MOS transistor is between 10-50, which is a fairly small number. In order to increase the gain, channel lengths can be increased to reduce $\lambda$ (suppress the channel length modulation effect). However, the achievable gain is still quite low. Also, as device size increases, the parasitic capacitance associated with the device also increases. Frequency response of the device will degrade. To attack this difficulty, cascoding and gain-boosting techniques can be used. Figure 5.2 shows four candidate topologies. Figure 5.2(a) and 5.2(b) are two simple topologies. They can increase the intrinsic gain by a factor $\approx g_m r_o$. But this still cannot meet the desired specification. Figure 5.2(c) and 5.2(d) use gain-boosting technique, which can significantly boost up the intrinsic gain by a factor of $A_v g_m r_{o3}$, where $A_v$ is the gain of the booster (i.e., the auxiliary amplifier). They should satisfy the gain requirement at the cost of area, complexity and more power consumption. These four amplifiers all have good speed, but due to the stack of the cascoding devices, they have very limited output swing. Thus the dynamic range is very small.



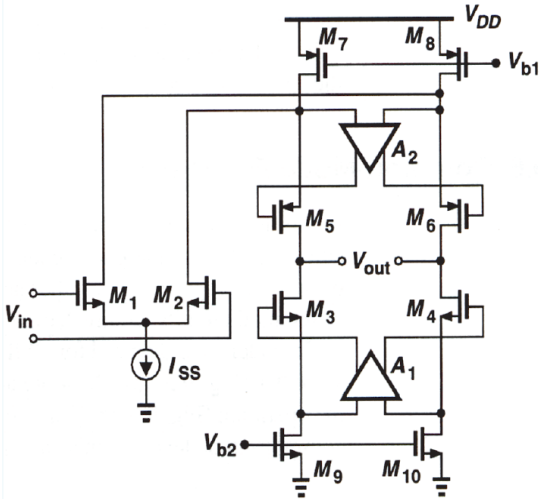

(a) Telescopic




(b) Folded-cascode




(c) Gain-boosted telescopic




(d) Gain-boosted folded-cascode

Figure 5.2: Four single stage amplifier topologies

To increase the open-loop gain,and at the mean time, provide a large output swing, a multi-stage topology may be employed. Although adding a third stage can improve the gain, the drawbacks are obvious: (1) it dissipates more power; (2) it deteriorates op amp frequency response because the 3rd stage introduces at least one more pole, which usually makes the op amp difficult to compensate and therefore deteriorate the overall frequency response. On the other hand, there are several advantages of a two-stage topology. Firstly, with appropriate design, two-stage configuration can well balance the gain and bandwidth tradeoff. Secondly, in a typical two-stage op amp, the noise is attenuated by the gain of the first stage when it's referred back to the inputs. Thus the noise of a two-stage amplifier is comparable to that of a single stage amplifier. Thirdly, the second stage or output stage can be designed to source and sink large currents (push-pull) to improve the slew rate. These benefits suggest that the tradeoffs among gain, noise, bandwidth and output swing can be significantly mitigated by employing a two-stage topology. It should be noted though, the traditional, simple two-stage topology is not sufficient due to its limited gain (below $70dB$) in the deep submicron regime. The cascoding structure was adopted in this design. The gain-boosting technique was not used, because the op amp appears in every CAB of the FPAA and the gain-boosting topology will add significant area cost and power consumption to the system.

**Input Stage** On the TSMC018 CM process, the supply voltage is $3.3V$. This provides a good voltage headroom. For the first stage, a topology that allows for high gain, low noise and low power consumption is desired. Here, output swing is less important since high swing can be obtained in the second stage. As discussed in

the previous section, the open-loop gain for regular two-stage amplifier is fairly small for deep submicron CMOS technology (below $70dB$). The gain-boosting technique should not be used, because it takes up more area and add significant amount of power consumption to the FPAA. To increase the gain, cascoding technique can be adopted. The available options are folded-cascode and telescopic structures (figure 5.2). Telescopic structure has slightly higher gain and better frequency response because the second dominant pole of the folded-cascode structure is closer to the origin. When further comparing these two topologies, it should be noticed that, to minimize power dissipation, the number of current legs in the amplifier must be minimized. This favors telescopic topology compared to folded cascode. Also, in a two-stage amplifier, noise is dominated by the first high gain stage. This means the input devices and the active loads will contribute significant amplifier noise. The folded cascode has more devices in the signal path, which contribute more noise. Therefore a telescopic first stage will be a better choice. In this work, a novel fully balanced, telescopic differential difference input stage is used as the first high gain stage.

**Output Stage** For the second Stage, the main concern in selecting the appropriate configuration is the output swing and its driving capability. In comparing the class A and class AB output stages, the latter allows for a smaller standby biasing current while still being able to source and sink large current for dynamic transitions. These advantages make class AB outperform class A configuration. In this design, a common-source class AB output stage was employed. It allows a nearly rail-to-rail output swing, i.e., one overdrive voltage within the supply rails, and low

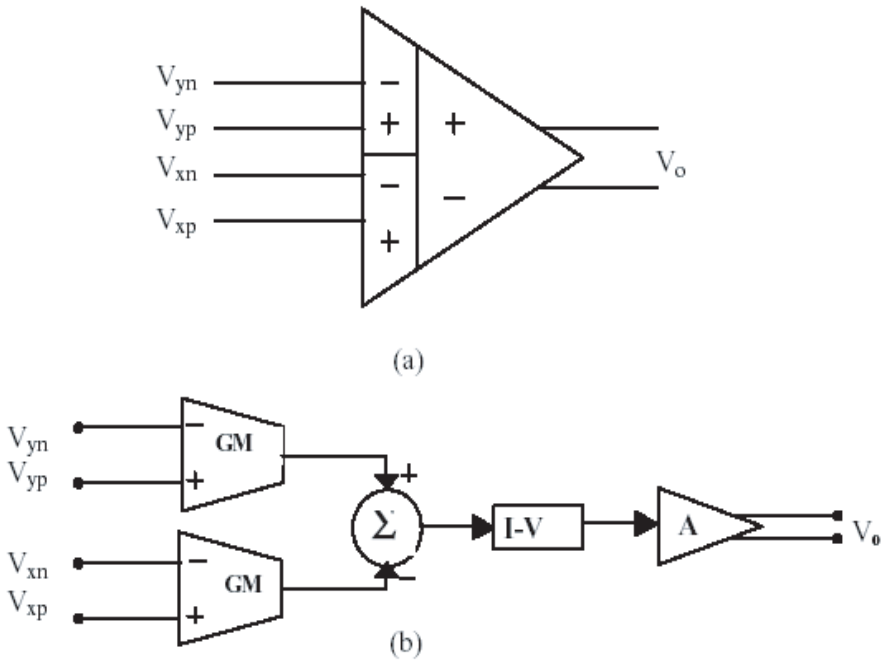



Figure 5.3: The DDA conceptual block diagram (a)symbol; (b)block diagram

power consumption.

## 5.2 Design of the Differential Difference Op Amp

In this work, a novel differential difference amplifier (DDA) was developed (figure 5.3). Using the notations similar to those in [75], the signal variables of DDA are described by four vectors:

$$\begin{pmatrix} V_{id} \\ V_{cy} \\ V_{cx} \\ V_{cd} \end{pmatrix} = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \end{pmatrix} \begin{pmatrix} V_{yp} \\ V_{yn} \\ V_{xp} \\ V_{xn} \end{pmatrix} \tag{5.2}$$

The differential voltage $V_{id} = (v_{yp} - v_{yn}) - (v_{xp} - v_{xn})$ is what needs to be amplified. The other three vector components are common-mode voltages and usually should

not be amplified. Ideally, the DDA amplifies the differential voltage $v_D$ by an near infinite amount and fully suppress all common-mode voltages:

$$v_o = A_0[(v_{yp} - v_{yn}) - (v_{xp} - v_{xn})] \tag{5.3}$$

where $A_0$ is the open-loop gain. When negative feedback is applied and A0 $\rightarrow \infty$, $v_{yp} - v_{yn} = v_{xp} - v_{xn}$. As the open-loop gain decreases, the difference between the two differential voltages increases. Therefore, the open-loop gain is required to be as large as possible in order to improve the performance.

The output of the non-ideal op amp with the parameters of its linear model can be characterized as:

$$\begin{aligned} V_o = A_d[(v_D - V_{off}) &+ \frac{1}{CMRR_y}(v_{cy} - V_{cy0}) \\ &+ \frac{1}{CMRR_x}(v_{cx} - V_{cx0}) \\ &+ \frac{1}{CMRR_d}(v_{cd} - V_{cd0})] \end{aligned} \tag{5.4}$$

where $A_d$ is the open-loop gain, $V_{off}$ is the offset voltage, $CMRR_y$ and $CMRR_x$ are Y port and X port common-mode rejection ratios. $CMRR_d$ is a new parameter that is not available for general two-input op amps. It measures the effect of equal floating voltages at the two input ports. The nonlinear function is linearized around the biasing points $v_o = V_{CM0}$, $v_{yp} = V_{yp0}$, $v_{xp} = V_{xp0}$ $v_{cd} = V_{cd0}$. This equation indicates that to improve the common-mode-rejection-ratio, not only each of the transistors in the X or Y port should be matched, the two differential pairs should also match each other (to improve $CMRR_{cd}$). Thus a well-planned layout arrangement is crucial to the amplifier's performance.

### Input Stage

Figure 5.4 shows the schematic of the fully differential input stage. NMOS transistors are faster than PMOS transistors due to the higher mobility of electrons than that of holes. As a result, amplifiers with all NMOS transistors on the signal paths will have higher speed than the their PMOS counterpart[1]. Therefore NMOS transistors were used in the two differential pairs. Transistor M11 an M12 are the tail current sources. Transistor M1 to M4 form the two differential pairs, which convert the differential input voltages into currents. The two differential pairs are drain cross-coupled. Transistors M21, M22 are cascoding devices. Cadcoded current sources were used as active loads, where M5 and M6 are cascoding current source loads. They convert the differential currents into two differential output voltages, which are the inputs to the second stage. Since the two outputs $V_{out1\_L}$ and $V_{out1\_R}$ are high impedance nodes, a little mismatch between the currents through M11 and M12 and current sources on the top through M21 and M22 will result a large voltage drift from the desired common-mode output voltages. Thus a common-mode feedback (CMFB) block must be employed. Here, transistor M5C and M6C are used to adjust the common-mode output voltage of the first stage. Their gates are controlled by the common-mode feedback signal. Voltages $V_{bp}$, $V_{b2}$, $V_{b1}$ and $V_{bn}$ are used to properly bias the telescopic stage. A supply independent, high swing cascoding biasing block was used to generate these voltages. This biasing block

[1]For some low noise applications, it would be beneficial to use PMOS transistors as the input pair because of their smaller 1/f noise. However, this should not be a major concern for this general purpose FPAA.

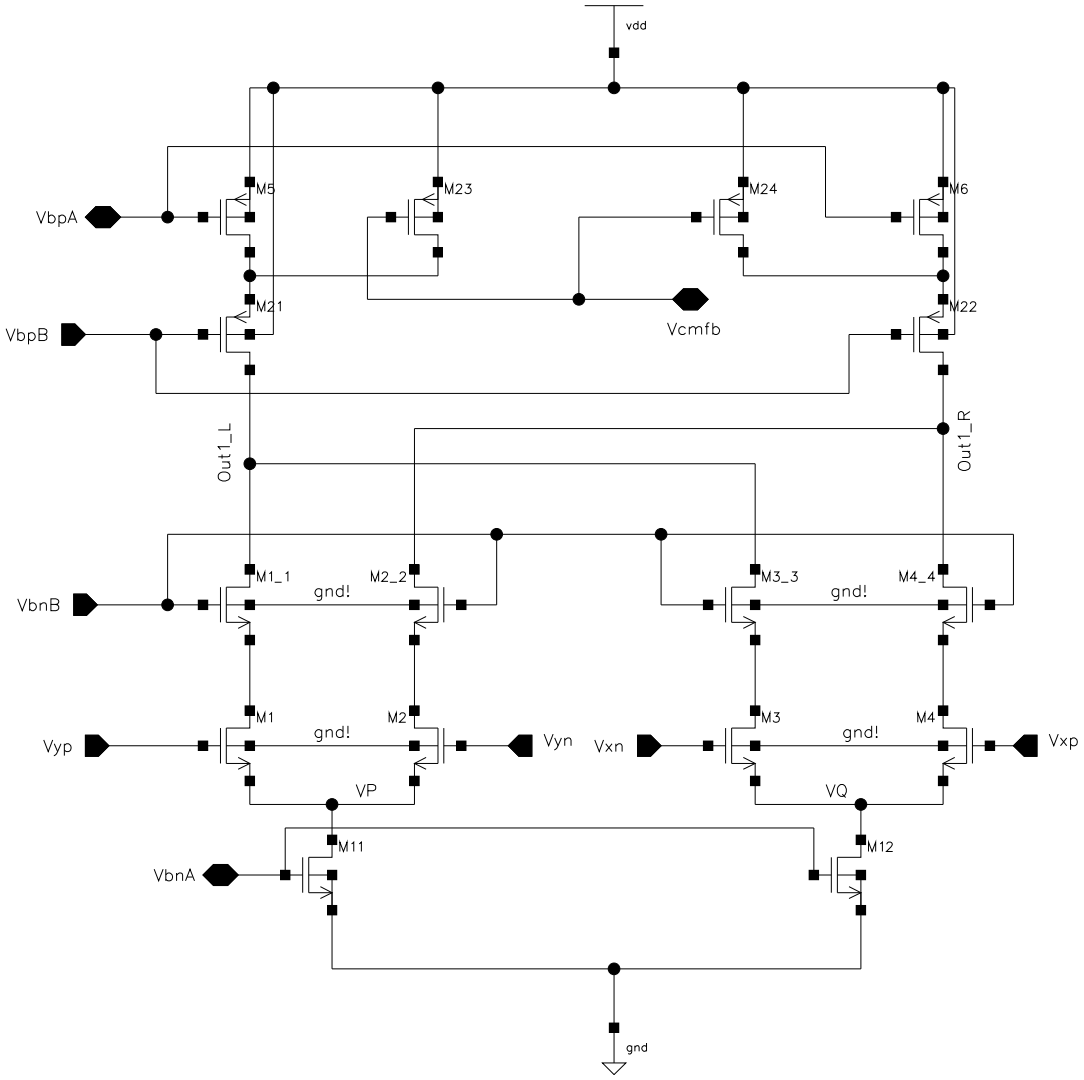



Figure 5.4: The input stage of the DDA

will be discussed later. One drawback of this input stage topology is its limited common-mode input range.

The outputs of the first stage can be expressed as:

$$V_{out1} = V_{out1\_L} - V_{out1\_R} = A_1[(V_{yp} - V_{yn}) - V_{xp} - V_{xn}] \tag{5.5}$$

where $A_1$ is the small signal voltage gain of the input stage, as shown below:

$$A_1 = g_{m1,2,3,4}(r_{o,up}//r_{o,down}) \tag{5.6}$$

$$r_{o,up} = r_{o21} + [1 + (g_{m21} + g_{mb21})r_{o21}](r_{o5}//r_{o5c}) \quad (5.7)$$

$$r_{o,down} = r_{o,down1}//r_{o,down2} \quad (5.8)$$

$$r_{o,down1} = r_{o1\_1} + [1 + (g_{m1\_1} + g_{mb1\_1})r_{o1\_1}]r_{o1} \quad (5.9)$$

$$r_{o,down2} = r_{o3\_3} + [1 + (g_{m3\_3} + g_{mb3\_3})r_{o3\_3}]r_{o3} \quad (5.10)$$

In the above equations, $g_{m,x}$, $g_{mb,x}$ and $r_{o,x}$ are the transconductance, body transconductance and output resistance of transistor x, respectively. The input stage incorporates cascoding devices (M5-M21, M1-M1_1 etc), with the two differential pairs. It takes advantages of the telescopic structure, namely high gain and excellent frequency response. This stage along will be able to provide DC gain of approximately $60dB$.

**Output Stage** The output stage is also a fully differential structure. It consists of two identical common-source class AB output stages (M7-M10 and M17-M20) [76], which can provide large output current diving capability with relatively low standby power consumption. The common-source topology ensures a large output swing, about one $V_{ov}$ within the supply rails. The problem, the common-mode output voltage of the first stage is not the same as the DC bias point required by the second stage input, is solved through the use of voltage level-shifting transistors M15 and M16. Transistors M15 and M16 are used to properly bias both stages. Since transistors M7 and M8 are set to have the same size, to the first order, they carry the same current, thus they have the same gate source voltages (i.e., $V_{gs7} = V_{gs8}$).

$$V_{sg9} + V_{sg7} + V_{sg10} = V_{DD} \quad (5.11)$$

$$V_{sg15} + V_{sg16} + V_{sg7} = V_{DD} \quad (5.12)$$

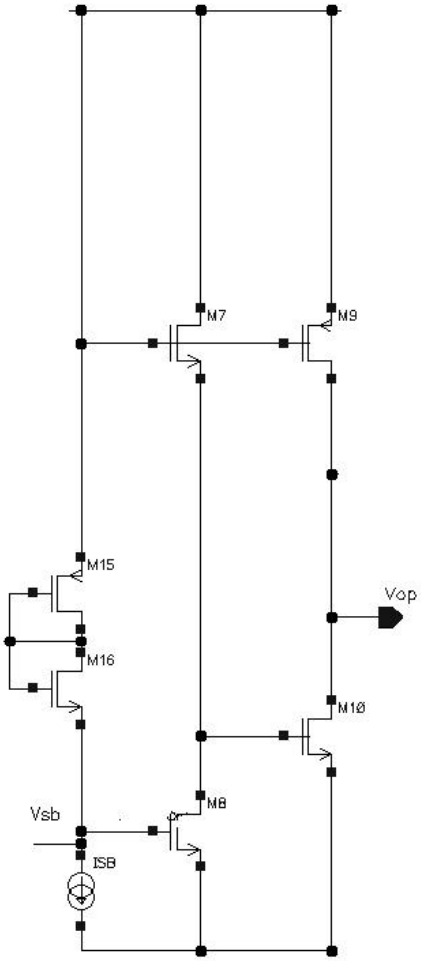



Figure 5.5: The output stage of the DDA

From equations (8) and (9), it can be derived that:

$$V_{sg9} + V_{sg10} = V_{sg15} + V_{sg16} \tag{5.13}$$

To the first order, the current through transistors M9 and M10 equals the current flowing through transistors M15 and M16. When the sizes of transistors M7, M8, M9 and M10 are properly defined, the gate voltage of transistor M10 would be slightly larger its threshold voltage. In other words, during standby mode, the

current through M9 and M10 is can be set by the current source $I_{SB}$, which is usually a very small static current. Thus the static/standby power consumption can be kept to minimum. However, when the circuit is in normal operation, either transistor M9 or M10 or both will work in saturation region, or one is fully turned on, another is turned off (depends on the signal level at the outputs of the first stage). The common-source configured transistors M9 and M10 will supply large current to drive the capacitive load.

The small signal gain of the second stage can be expressed as:

$$A_2 = (g_{m9} + g_{m10})(r_{o9}//r_{o10}) \tag{5.14}$$

In order to properly set the output common-mode voltage, the output voltages are sensed and compared with the reference voltage. The generated control voltage is fed back to adjust the output voltages of the first stage. Consequently, the common-mode voltages of the second stage are adjusted.

**Compensation for the Op Amp**

Compensation is required to maintain stability of most amplifiers when they are configured in some form of feedback loop. For this two-stage op amp, the dominant pole is at node $V_{out1}$, and the first nondominant pole is at the final output node $V_{out}$. The traditional Miller compensation scheme places a pole-splitting capacitor between the final output of the amplifier and the output of the first stage of the amplifier. This has the effect of creating a low frequency, dominant pole and moving the second pole to a higher frequency which will ensure amplifier stability. Due to the relatively small $g_m$ of MOS transistors, a right half place (RHP) zero is closed to

the non-dominant poles [75]. This brings stability problem. Several methods have been developed to eliminate this undesired RHP zero such as adding a follower stage at the cost of more power consumption and complexity. A simpler method would be using a nulling resistor (which is usually implemented by a MOS transistor) to cancel the zero. However, this requires an extra biasing voltage to adjust the effective resistance of the nulling transistor. Thus, an alternative method, cascode compensation scheme was used in this design. It creates a dominant pole and two complex poles at higher frequency by placing a compensation capacitor between the amplifier output and first stage cascode node. This will also ensure amplifier stability when it is placed in a feedback loop. Although both compensation schemes ensure stability, the cascode compensation scheme improves the speed of the amplifier as compared to the standard Miller compensation method [77].

**Common-Mode-Feedback (CMFB) Block** The current of the current source loads on the top is essential set by the gate biasing, while the current at the bottom half circuit is set by the tail current source. Because the high impedance of the cascoded structure, a slightly current mismatch will result in large common-mode variation. Therefore, CMFB block is necessary for all the fully differential amplifier with active loads. Switched-capacitor based CMFB scheme was not considered here because the op amp was designed to operate in continuous-mode. Figure 5.6 is a widely used scheme that uses transistors only[78], [79]. This scheme does not resistively load the op amp outputs, but the source-coupled pairs $M_{C11}$ and $M_{C22}$ capacitively load the op amp outputs. More importantly, the proper operation of this CMFB block requires $M_{C11}$, $M_{C12}$, $M_{C21}$ and $M_{C22}$ to remain on during the

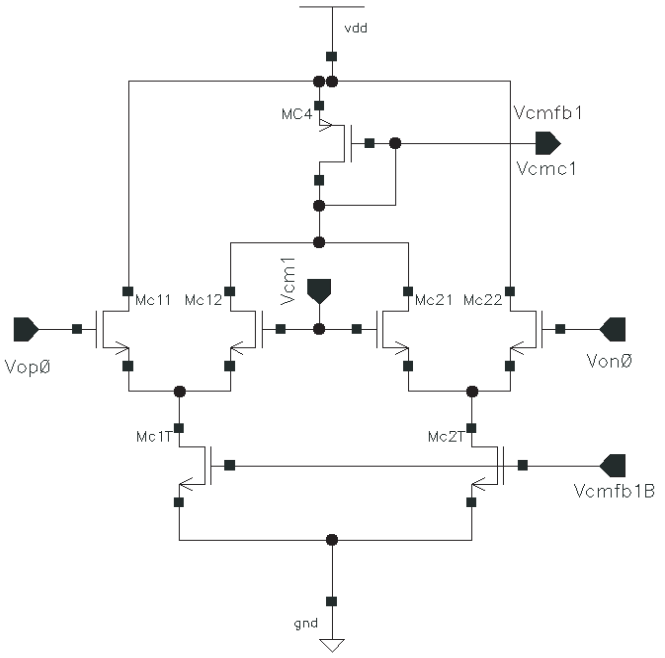



Figure 5.6: A transistors-only common-mode feedback circuit

entire output swing. As a result, the output swing is limited. Since dynamic range is an important parameter for this op amp, the CMFB block shown in figure 5.7 was used. The common-mode output voltage $V_{oc} = \frac{V_{op}+V_{on}}{2}$ is sensed by the resistors. And this value is compared with the desired common-mode reference voltage $V_{cmrf}$. The amplified difference is sent back to control the gate bias in the first stage, which in turn adjust the output CM voltage until it's equal to $V_{cmrf}$. The CM sensing resistors with the input capacitance of the CM sense amplifier differential pair will introduce a pole in the CMFB loop. This degrade the CMFB loop gain at higher frequency. The two capacitors in parallel with the resistors are used to introduce a left-half-plane zero to slow down the gain drop, so CMFB still functions at fairly high frequency. Although this scheme may resistively load the amplifier, the gain is already high enough to meet the requirement.

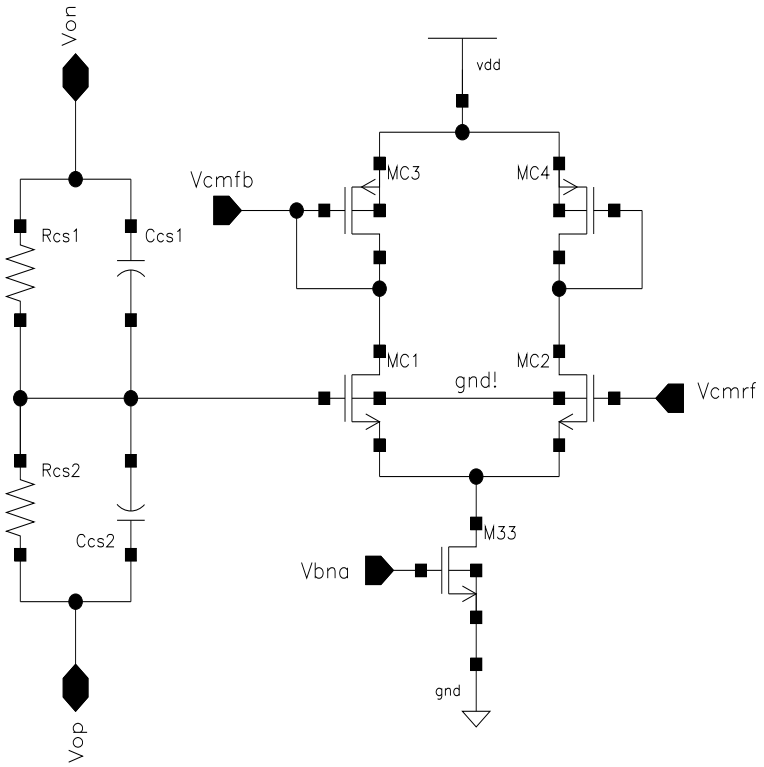



Figure 5.7: The common-mode feedback circuit used in this design

### The Supply Independent, High-Swing Biasing Block

The Supply Independent Biasing Block In an op amp and many other analog circuits it is usually desirable that the quiescent conditions be stabilized with respect to variations in the supply voltage. Reducing the sensitivity of current sources to changes in the supply voltage is essential to making circuits immune to noise on the supply. Generally, the current source is directly derived from the power supply $V_{DD}$. To reduce the sensitivity of the current source to $V_{DD}$, a voltage value other than power supply must be used. In this work, a $V_{th}$ referenced supply independent biasing circuit was used, as shown in figure 5.8. The current is set by $V_{GS1}/R$, which suggests the current is independent of power supply. However, due to MOS transistors finite output resistance, it current will still be related to $V_{DD}$ but at much

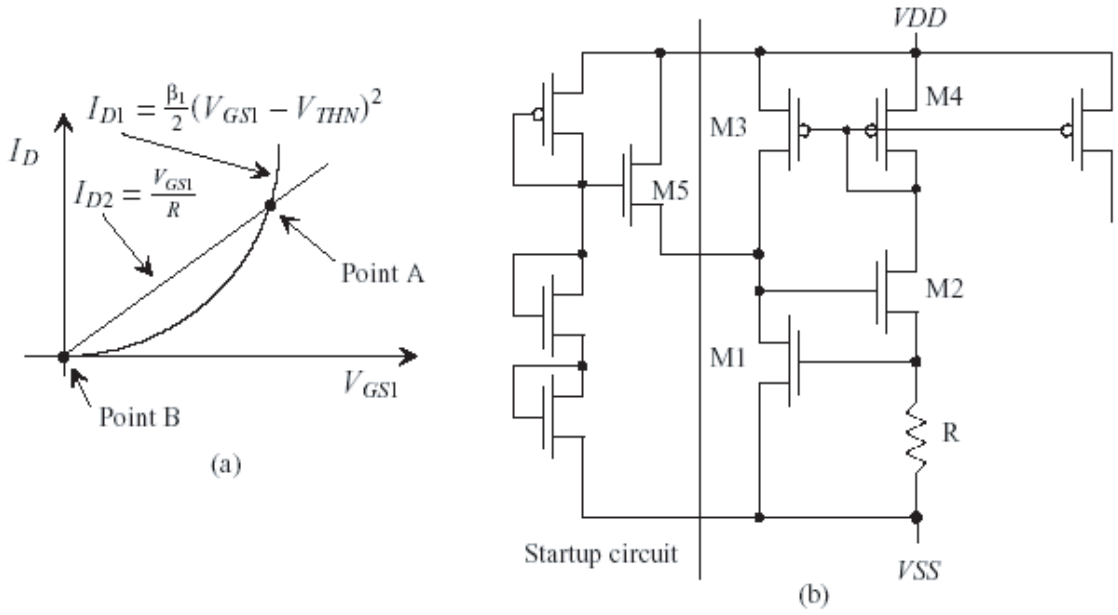



Figure 5.8: The $V_{th}$ referenced biasing block (a) two possible operating points (b) the complete biasing block with a startup circuit.

less sensitive level comparing the current directly set by the power supply. It should be noted that there are two possible operating points, as shown in figure 5.8 (a). Point B, where only leakage currents flow, should be normally be unstable. However, in practical circuits the transistor current gains degrade at very low currents. As a result, B may be a stable operating point. Therefore, a start-up circuit was used to ensure that $I_{D2} \geq 0$. When the circuit is stuck at point B, transistor $M_5$ is used to pull up the gate voltage of $M_2$ until the circuit goes back to the normal operation point A. At this point, the gate-source voltage of $M_5$ is much smaller than the threshold (due to the two stacked diode-connected NMOS transistors on the left) and it turns off.

The current source $I_{D2}$ is given as:

$$I_{D2} = \frac{V_{GS1}}{R} \tag{5.15}$$

Since $V_{GS1} = V_{th1} + V_{ov1}$, the temperature dependence of the current source is:

$$\begin{aligned} TC_{I_{ref}} &= \frac{1}{I_{D2}}\frac{\partial I_{D2}}{\partial T} \\ &= \frac{1}{V_{GS1}}(\frac{\partial V_{th1}}{\partial T} + \frac{\partial V_{ov1}}{\partial T}) - \frac{1}{R}\frac{\partial R}{\partial T} \\ &= \frac{V_{th1}}{V_{GS1}}TC_{V_{th1}} + \frac{V_{ov1}}{V_{GS1}}TC_{V_{ov1}} - TC_R \end{aligned} \tag{5.16}$$

It's well known that MOS transistor threshold voltage has a negative $TC$ [75]. The overdrive voltage also has a negative $TC$ which primarily is due to the negative temperature dependence of the electron/hole mobility. By properly choosing a type of resistor with a certain amount of negative $TC$ on this specific CMOS process, the temperature coefficient of the source current will be minimized. The simulation result shows a overall $TC_{I_{ref}}$ of approximately 240ppm/$^\circ C$.

The High Swing Biasing Block The cascoded current source is a useful structure that is widely used as active load in many analog circuits. The easiest way to bias the cascoded current mirrors is shown in figure 5.9:

To keep transistors $M_3$ and $M_2$ in saturation, the minimum voltage of nodes $P$ and $Y$ should satisfy:

$$V_{P,min} = V_{GS1} + V_{GS0} - V_{th3} \approx V_{th} + 2V_{ov} \tag{5.17}$$

$$V_Y = V_{GS1} + V_{GS0} - V_{GS3} \approx V_{th} + V_{ov} \tag{5.18}$$

This means to keep both $M_3$ and $M_2$ in saturation there's a high voltage overhead, because ideally only $2V_{ov}$ is needed. This $V_P$ voltage will limit the amplifier swing in

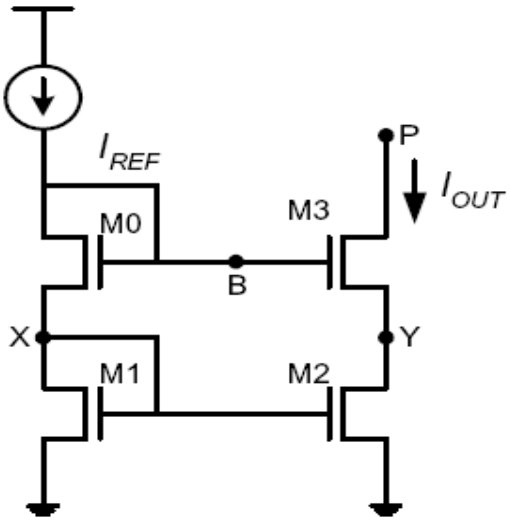



Figure 5.9: Biasing the cascoded current mirror

the first stage. To reduce the overhead, the voltage at node $Y$ should be minimized to a value slightly higher than the $V_{ov}$. A high swing biasing block called Sooch cascode current mirror similar to figure 5.10 [75] was used in this design. Transistor $M_5$ is deliberately set to operate in the triode region. If all the transistors have the same aspect ratio $W/L$, then when $M_5$ is sized as $\frac{1}{3}\frac{W}{L}$, the gate voltage would be $V_{th}+V_{ov}$. The drain voltage of $M_1$ is about $V_{ov}$. As a result, one $V_{th}$ voltage is saved for the swing. The transistor $M_4$ here is also operated in saturation region. This ensures $M_3$ and $M_1$ have the same drain-source voltage and improve the current matching. In the actual design, the aspect ratio of $M_5$ was smaller than $\frac{1}{3}\frac{W}{L}$ to leave some room to make $M_1$ and $M_2$ stay in saturation region.

**The Complete DDA Circuit** The complete amplifier schematic including the amplifier core and the whole biasing block is shown in figure 11 and 12. When designing this amplifier, the first thing determined was the tail current of

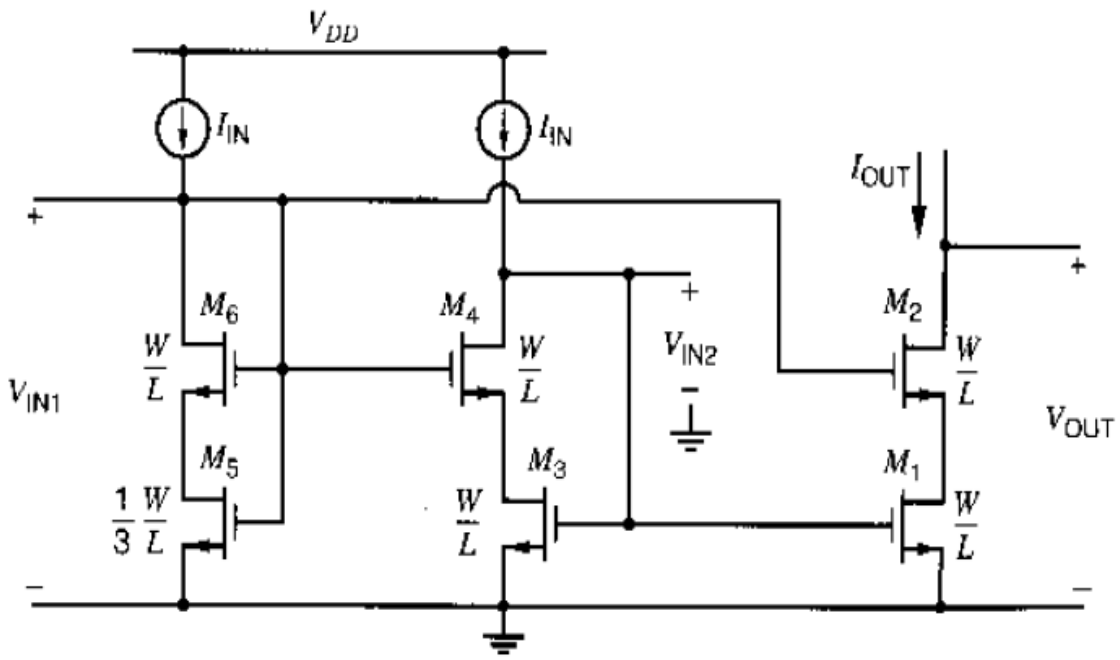



Figure 5.10: The high swing Biasing block

the two differential pairs. Based on previous design experience, the compensation capacitance is in the range of a few hundred femto farads to a pico farad. To achieve higher than $100V/us$ slew rate, approximately $100uA$ current was picked for each tail current source. Because the two differential pairs have drain cross coupled, this guarantees at least $200V/us$ slew rate[2]. This relatively high biasing current also improves the bandwidth and settling time of the amplifier. As for the bandwidth concern, considering the worst case scenario with compensation capacitance of $1pF$, when the amplifier is Miller compensated with phase margin $PM \geq 45°$, there's only one dominant pole. Thus it can be modeled as a first order system with unity

[2]If the capacitance at the amplifier's final output is comparable or larger than the compensation capacitance, it can be charged or discharged through the class AB output stage, which essentially does not have slew limitation.

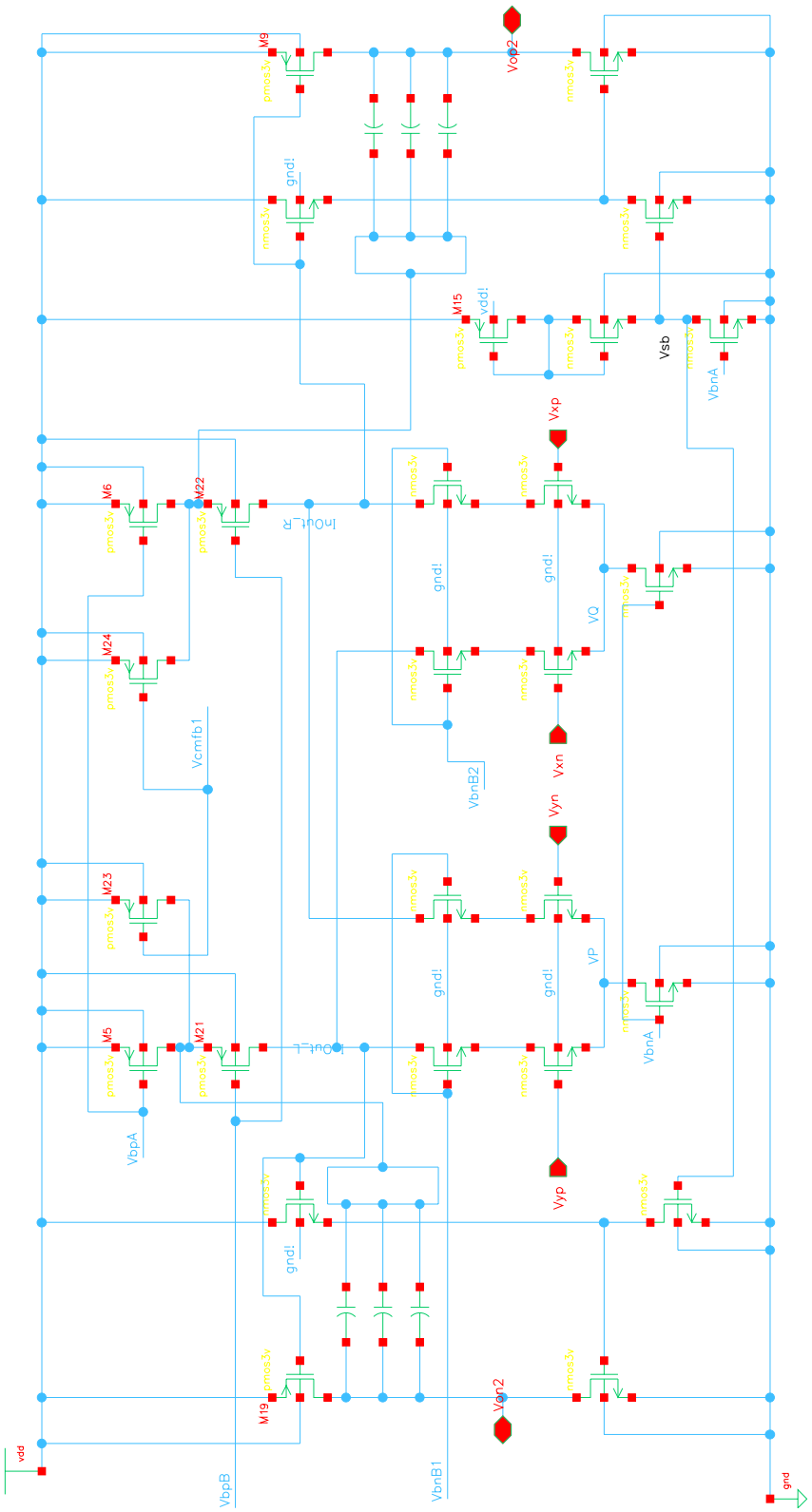



Figure 5.11: The amplifier core

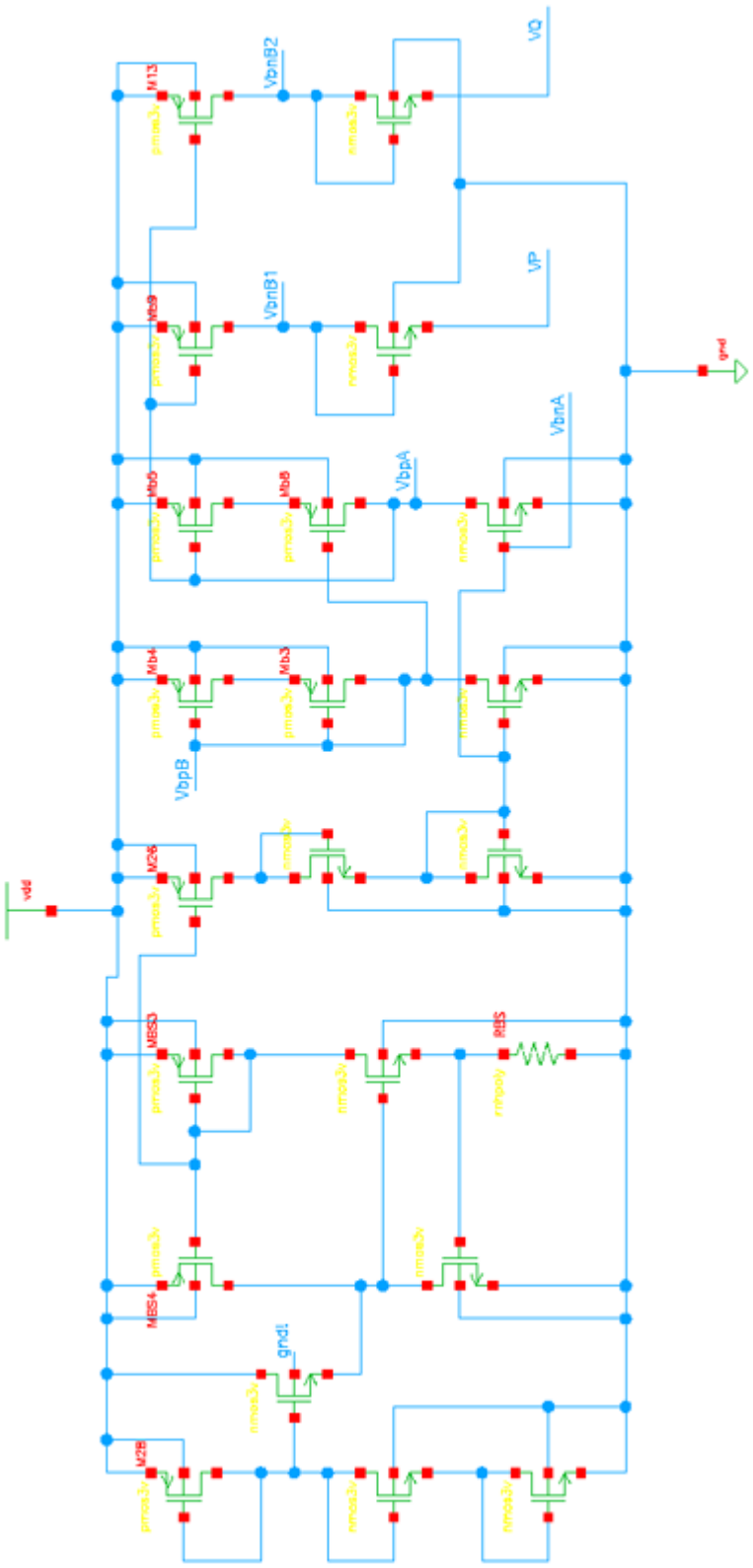



Figure 5.12: The complete biasing block

gain bandwidth (or gain-bandwidth product $GBW$) of $g_m/(2\pi C_c) = I_{tail}/(V_{ov}2\pi C_c)$. Put in the numbers and the estimated bandwidth is about $110MHz$, which certainly meets the desired specification.

Using laser Makelink, the first stage may be used along as a single stage amplifier or OTA (because there's only one high impedance node at its output). To get an adequate swing headroom, the sum of the overdirve voltages of the five stacked transistors were chosen to be half of the supply voltage. The four PMOS transistors $M_5$,$M_6$,$M_{23}$ and $M_{24}$ were allocated with higher overdrive $\approx 400mV$. They also have longer channel length. These two arrangements improve the overall amplifier performance because (1) both threshold voltage and transconductance parameter mismatches are inversely proportional to the square root of the transistor area [80]. Longer channel length reduces mismatch between current mirrors, and large overdrive minimize the effect of the mismatch; (2) these four PMOS transistors are the major noise contributors other than the four input devices. Since they do not capacitively load the signal path, increasing their size will reduce the noise without affecting the amplifier bandwidth. The four NMOS cascoding devices and the four PMOS cascoding devices directly contribute to the dominant pole of the amplifier, so they have shorter channel length. Since the speed/bandwidth is a major concern, the two differential pairs also have smaller channel length at the cost of higher offset and noise. The transistors in the output stage also have shorter length but large aspect ration. This improve the amplifier's frequency response.

**The Nonidealities and Layout Considerations** For any fully differential topology, the mismatch between transistors always has critical effect on the circuit

performance, especially on CMRR and offset voltage. The mismatch between the transistors inside the same differential pair has been analyzed extensively in many texts [75]. The emphasis here is focused on the mismatch effect between the differential pairs, $CMRR_d$. For simplicity, it's assumed the transistor sizes of the same differential pair are nominally matched.

Using the notation in equation (5.2), equation (5.2) can be written in a more general form:

$$v_o = A_{V2}[f_y(v_{yp} - v_{yn}) - f_x(v_{xp} - v_{xn})] \tag{5.19}$$

or:

$$v_o = A_{V2}[f_y(v_{cd} - \frac{v_{id}}{2}) - f_x(v_{cd} - \frac{v_{id}}{2})] \tag{5.20}$$

Here $A_{V2}$ is the gain of the second stage, $I = f(x)$ represents the voltage to current transfer function of the input stage times the output resistance. Ideally, when $v_{id}$ is zero, the output should be zero (AC ground).

The current $I_d$ in each differential branch satisfies:

$$I_d = \begin{cases} -I_{tail} & \text{for } V_d \leq -\sqrt{2I_{tail}/\beta} \\ \frac{1}{2}\beta\sqrt{\frac{4I_{tail}}{\beta} - V_d^2} & \text{for } |V_d| \leq \sqrt{2I_{tail}/\beta} \\ I_{tail} & \text{for } V_d \geq \sqrt{2I_{tail}/\beta} \end{cases} \tag{5.21}$$

Combining equation (5.20) and (5.21), the first stage differential gain $A_{V1}$ is:

$$A_{V1} = \frac{\partial v_o}{\partial v_{id}}|_{v_{cd}=V_{CD0}, v_{id}\to 0} \tag{5.22}$$

$$= A_{V2}\sqrt{\beta I_d}\frac{1-\frac{\beta}{I_d}\frac{V_{CD0}^2}{2}}{\sqrt{1-\frac{\beta}{I_d}\frac{V_{CD0}^2}{2}}} \tag{5.23}$$

where $\beta$ is MOS transistor transconductance. The gain is highest when the common-mode voltage of the two differential voltages $V_{CD0}$ is zero. If $\beta_X$ and $\beta_Y$ are the transconductance of the transistor within $X$ and $Y$ ports, and they are not identical, then according to the definition, $CMRR = A_{dm}/A_{cm-dm}$ ($A_{dm}$ is the differential gain, while $A_{dm\_cm}$ is the common-mode to differential gain) with some approximation:

$$CMRR_{d,stage1} \approx \frac{1}{1-\sqrt{\beta_Y/\beta_X}} \tag{5.24}$$

and the offset voltage between the two ports is:

$$V_{off} = \sqrt{\beta_Y/\beta_X - 1}\cdot V_{CD0} \tag{5.25}$$

Similarly, the second stage $CMRR$ ratio can be derived as:

$$CMRR_{d,stage2} \approx \frac{1}{1-\frac{\sqrt{\beta_{N'}}+\sqrt{\beta_{P'}}}{\sqrt{\beta_N}+\sqrt{\beta_P}}} \tag{5.26}$$

So the overall $CMRR$ of this op amp is:

$$CMRR_d \approx \frac{1}{1-\sqrt{\beta_Y/\beta_X}}\cdot\frac{1}{1-\frac{\sqrt{\beta_{N'}}+\sqrt{\beta_{P'}}}{\sqrt{\beta_N}+\sqrt{\beta_P}}} \tag{5.27}$$

The above equations show that $\beta$ mismatch reduces common mode rejection ratio and increases the offset voltage. Since there are two tail currents in the first stage, the mismatch between them also has negative effect. The $CMRR$ ratio and offset of the first stage due to tail current mismatch are:

$$CMRR_{d1} \approx \frac{1}{1-\frac{I_{tail1}}{I_{tail2}}}\cdot\frac{(2-\frac{\beta}{I_d}V_{CD0}^2)^2}{2+\frac{\beta}{I_d}V_{CD0}^2} \tag{5.28}$$

$$V_{off} \approx (\frac{I_{tail1}}{I_{tail2}} - 1) \cdot \frac{V_{CD0}}{2 - \frac{\beta}{I_d} V_{CD0}^2} \tag{5.29}$$

These errors could be due to (1) lithography and etching induced geometry mismatch, which result in device size deviation from the ideal value; (2) process induced mismatches which result in variations of the threshold voltage, gate oxide thickness and carrier mobility etc. Because the amplifiers exist in a fixed, pre-defined array based architecture, crosstalk and other noise sources are inevitable. Thus to minimize these negative effects and improve $CMRR$ is crucial to the overall system performance. This can be achieved by careful layout design.

The critical matching components include the four NMOS transistors of the two differential pairs and the two tail current sources. Each of the four transistors in the differential pairs was splitted into four parts and arranged as

$$\begin{bmatrix} A & B & C & D & D & C & B & A \\ D & C & B & A & A & B & C & D \end{bmatrix}$$

The two tail current sources also use the interdigitated structures and have the similar common-centroid arrangement. Dummy devices were used on the sides of the block to ensure the active devices have the same percentage of over/under etching. A plenty of substrate contacts were placed around the block to ensure device have even ground potential. They also function as a guardring to reduce the substrate coupling. The above layout techniques help to average out the process variations across the area and improve the geometry matching. Figure 5.13 and 14 show the layout of the input and output, respectively. The same strategy was extensively in the amplifier layout. Figure 5.15 is the final layout of the complete amplifier.

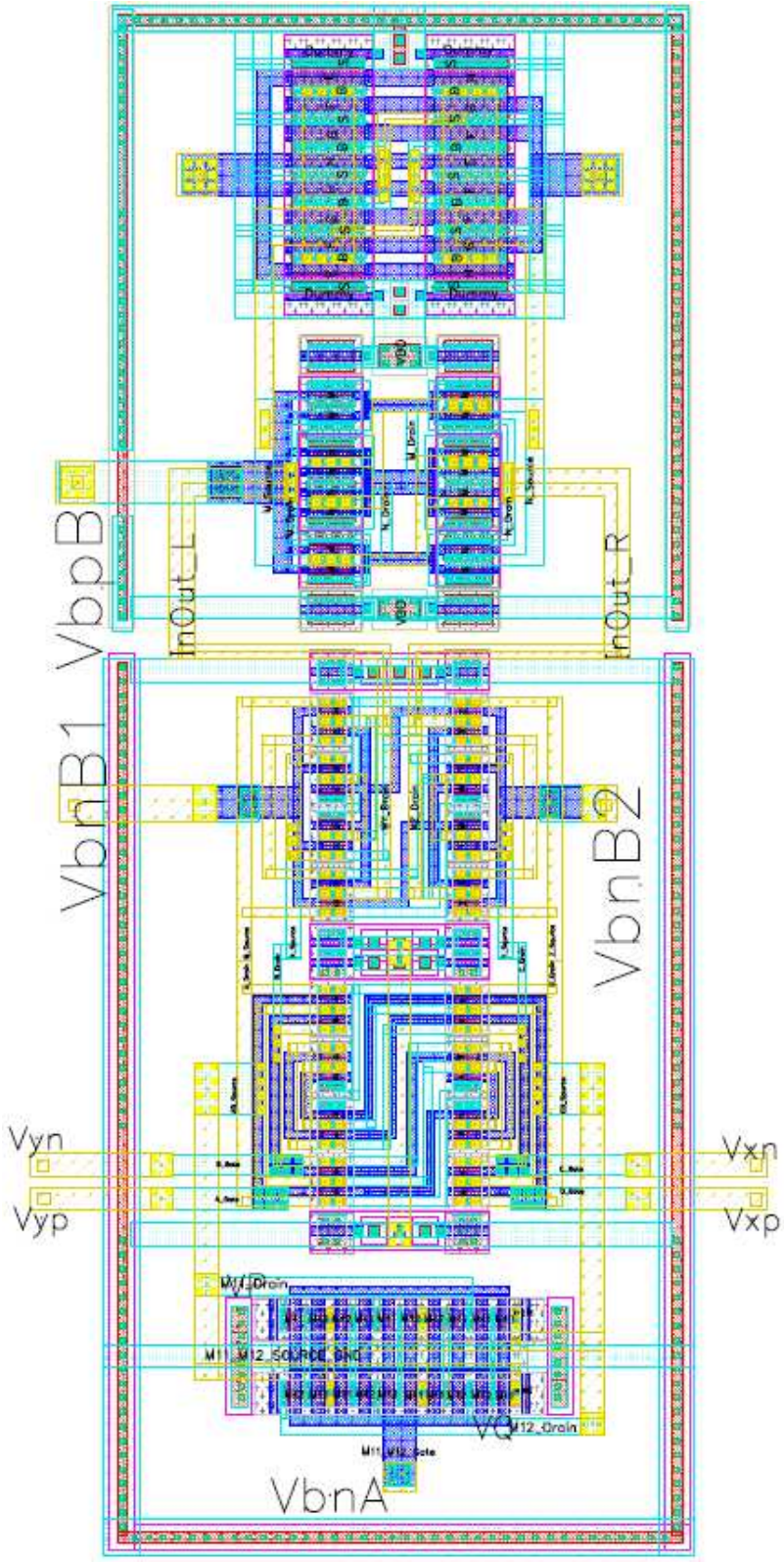



Figure 5.13: The fully differential input stage

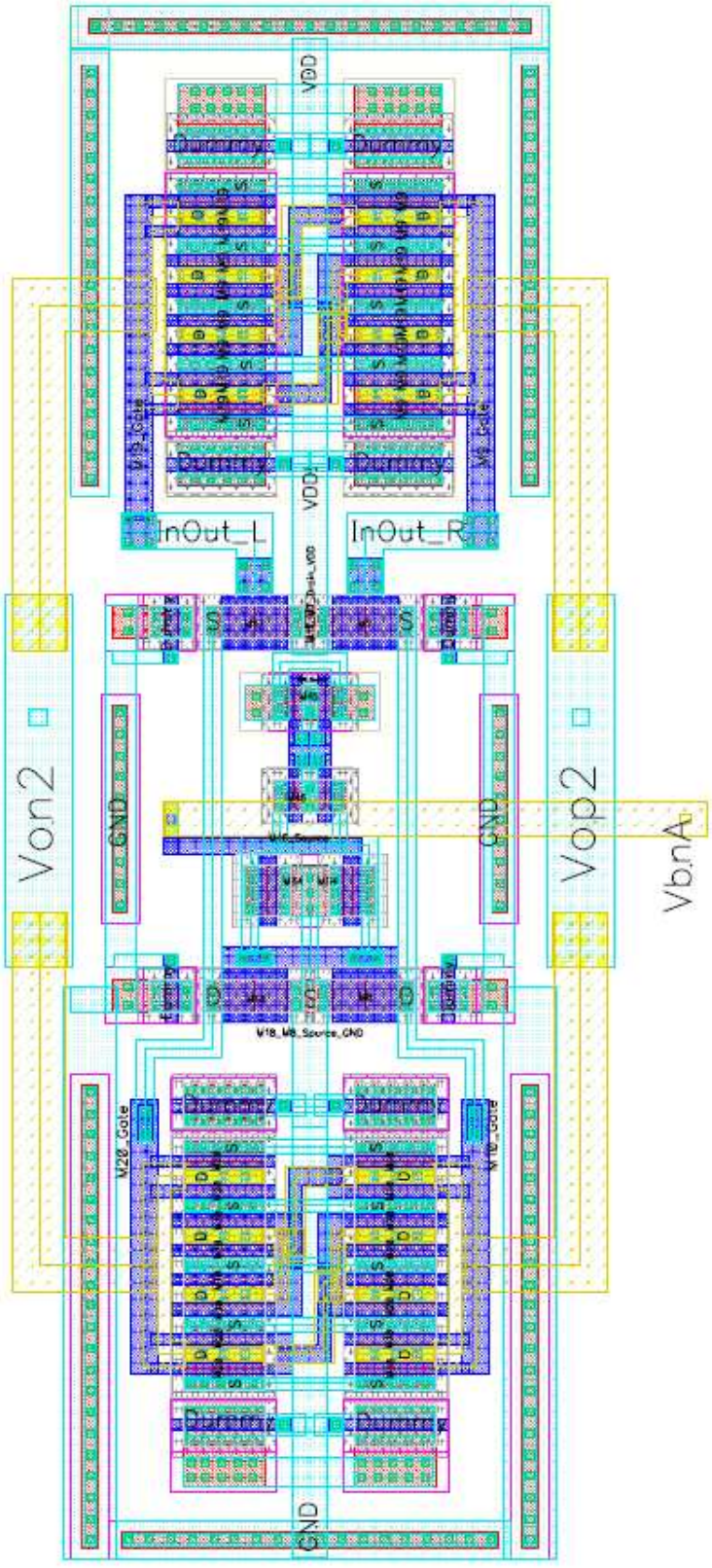



Figure 5.14: The class AB output stage

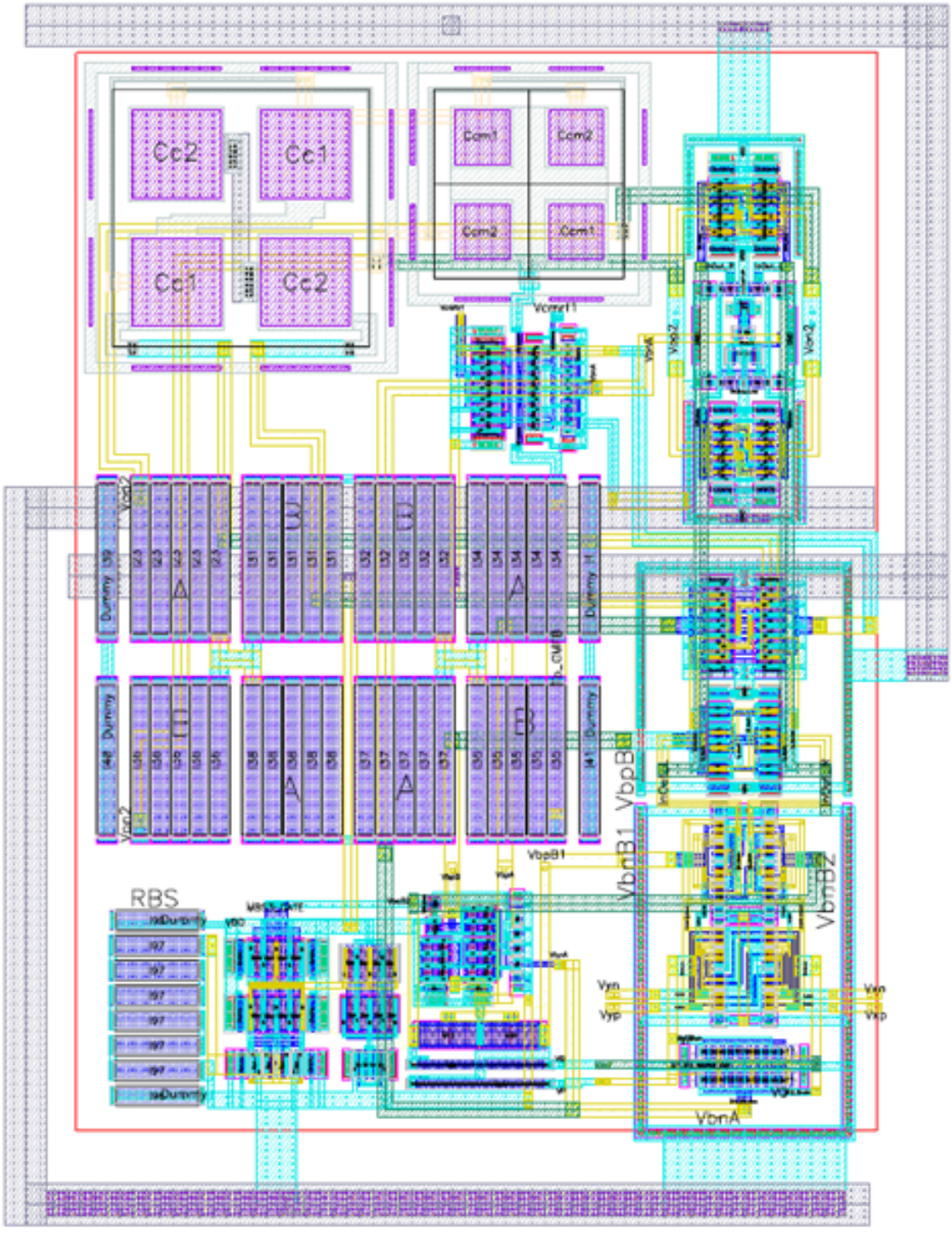



Figure 5.15: The complete amplifier layout

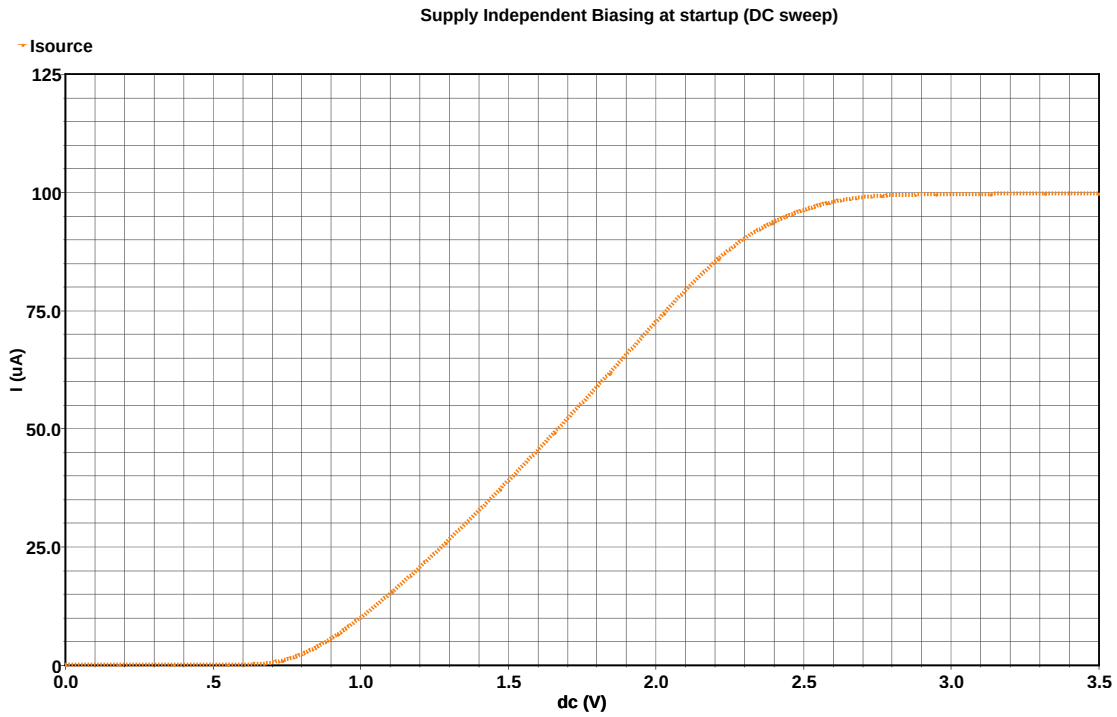
## 5.3 Results and Discussion

As mentioned previously, the problem of start-up of the supply independent biasing block must be carefully analyzed. The supply voltage was DC swept from zero to $V_{DD}$ (to exclude the parasitic capacitance caused false start-up) as well as in a transient test [81], as shown in figure 5.16. Both simulation and experimental results prove the proper start-up of this block.

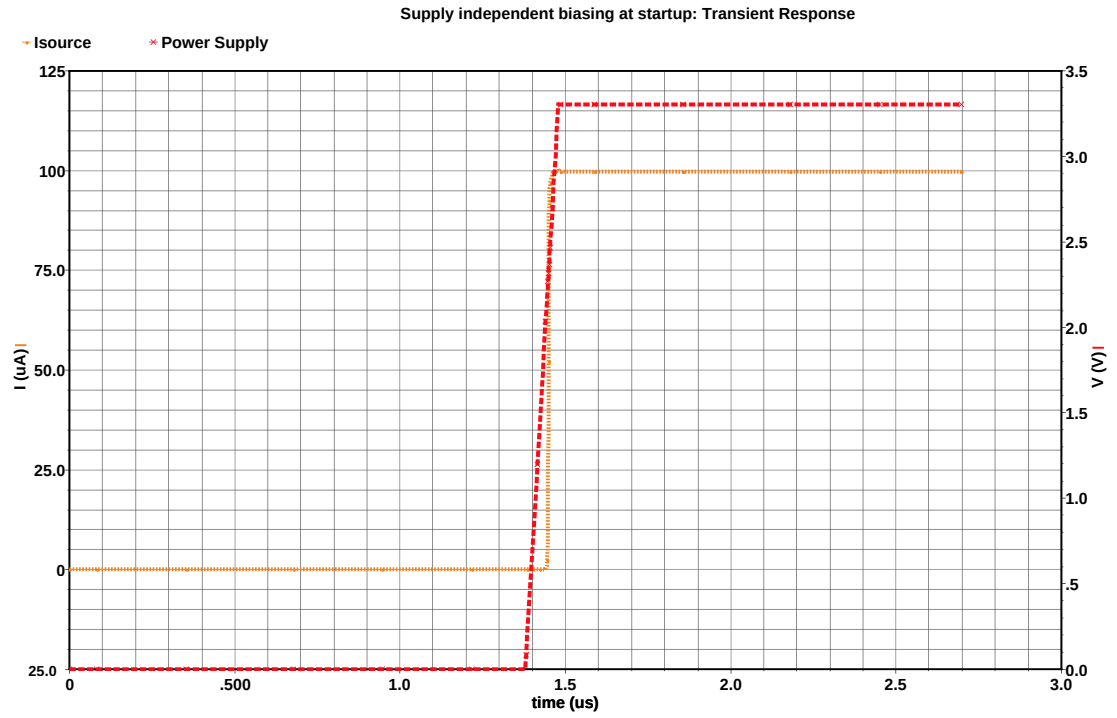
Figure 5.17 shows the biasing current as a function of the temperature. By the careful selection of a $P^+$ poly resistor with an appropriate negative $TC$, $TC$ of this current source is about $236ppm/^\circ C$, which is about an order of magnitude better than the general specification ($\approx 2000ppm/^\circ C$ [81]).

The open-loop frequency response (parasitic extracted post-layout simulation) is shown in figure 5.18. The unity gain bandwidth is well above $100MHz$ at $641.2MHz$ with $62^\circ$ phase margin with the nominal on-chip capacitance. Even with $1.5pF$ capacitive load, it still provides adequate phase margin of $45^\circ$.

When the devices are nominally matched, the $CMRR$ and $PSRR$ are extremely high for this fully differential amplifier. But this is an unrealistic condition which will never occur. In practice the actual rejection ratio is always measured, either explicitly or inexplicitly, with certain offset present. Figure 5.19 is the $CMRR_d$ as a function of frequency. Clearly the offset voltage has big impact on this performance figure of merit. On the contrary, the amplifier still shows a very good power supply rejection ratio $PSRR^+$ even with a relatively large offset of $15mV$. This benefits from the cascoded compensation scheme (cascoding structure and less




(a)




(b)

Figure 5.16: Supply independent biasing block at start-up (a) DC sweep; (b) transient

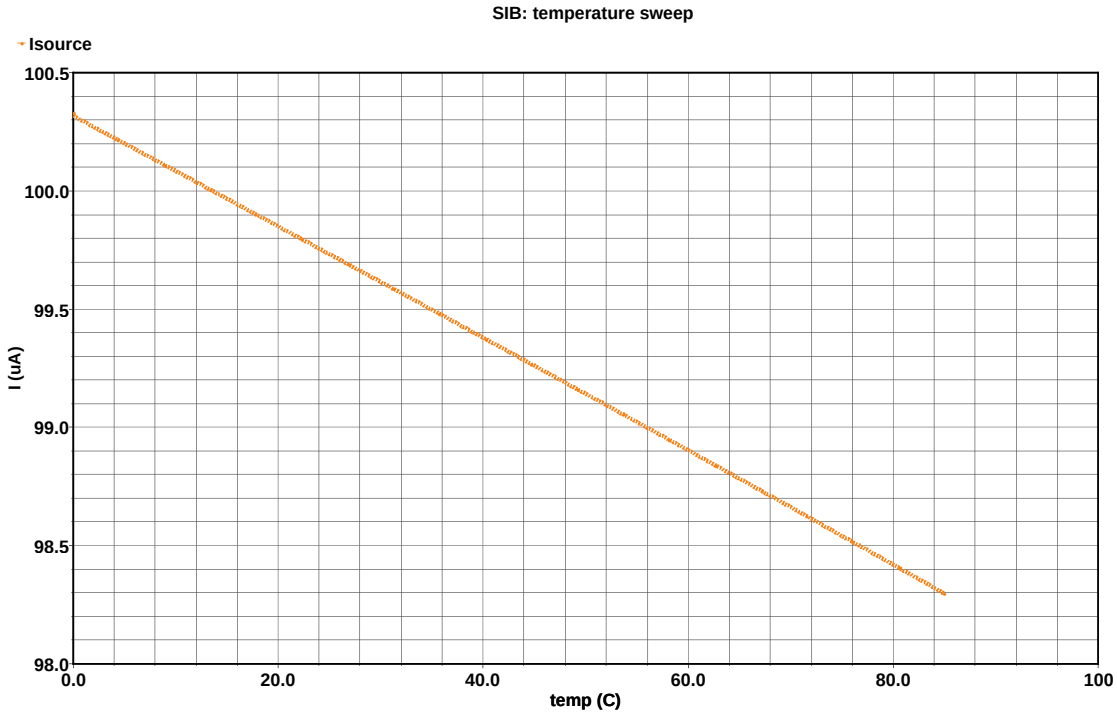



Figure 5.17: Temperature sweep of the supply independent biasing block

noise feeding through the $C_c$ and the supply insensitive biasing) as introduced in the previous section.

The amplifier's large signal and small signal transient response should also be carefully examined. They are important specifications for op amps used in data converters. Figure 5.21 and 5.22 are the amplifier's large signal and small signal step response, respectively. With typical on-chip capacitive load, the slew rate $SR$ is about $723V/uS$, and the settling time within 0.1% accuracy is $8.8ns$.

The amplifier can be configured to implement different gain by using a proper feedback. Figure 23 demonstrate the gain of 1, 2, 4, 8 and 16 as a function of frequency.

Finally figure 5.24 shows the input referred noise of this amplifier. At $1KHz$,

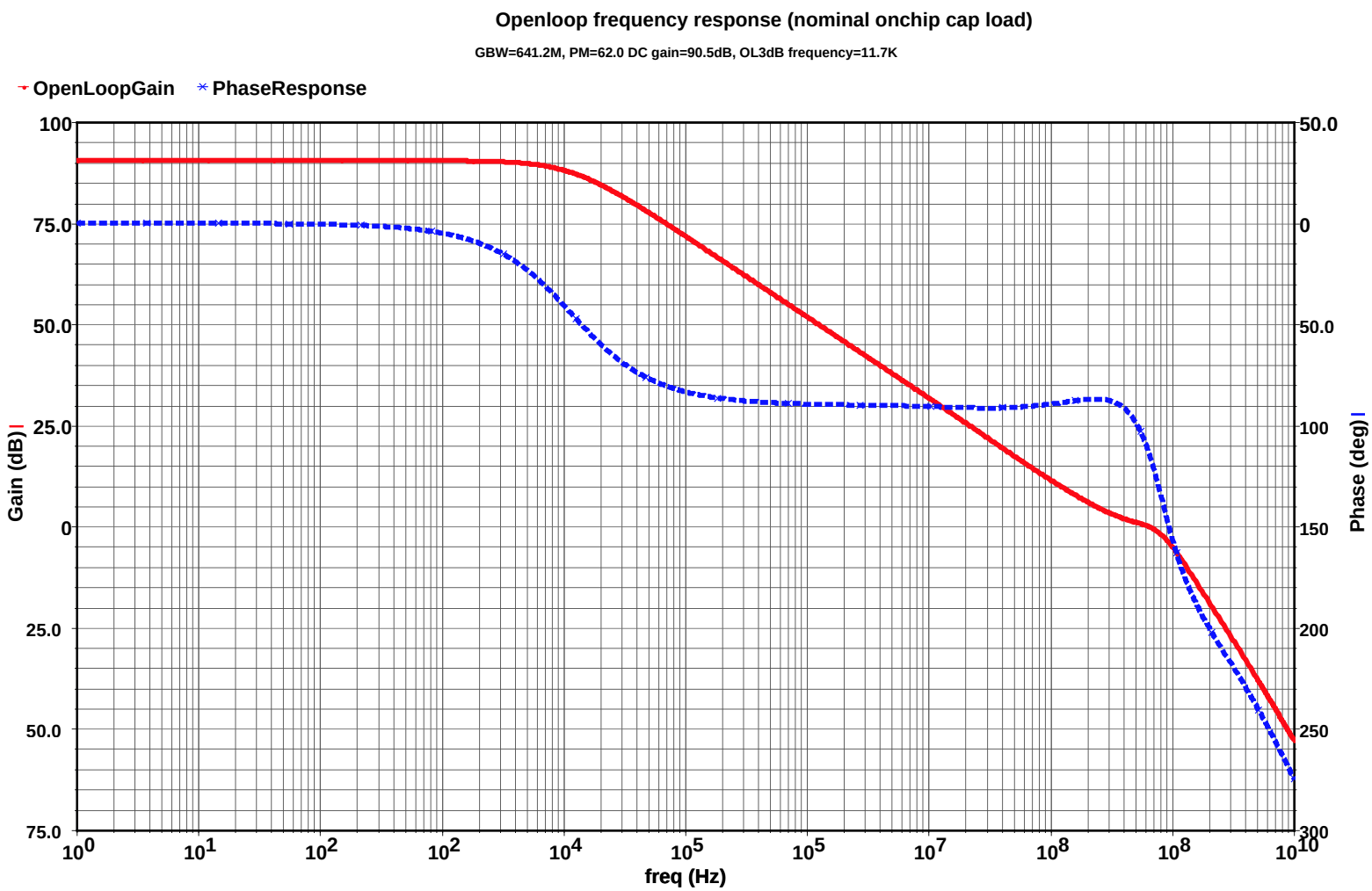



Figure 5.18: Open-loop frequency response

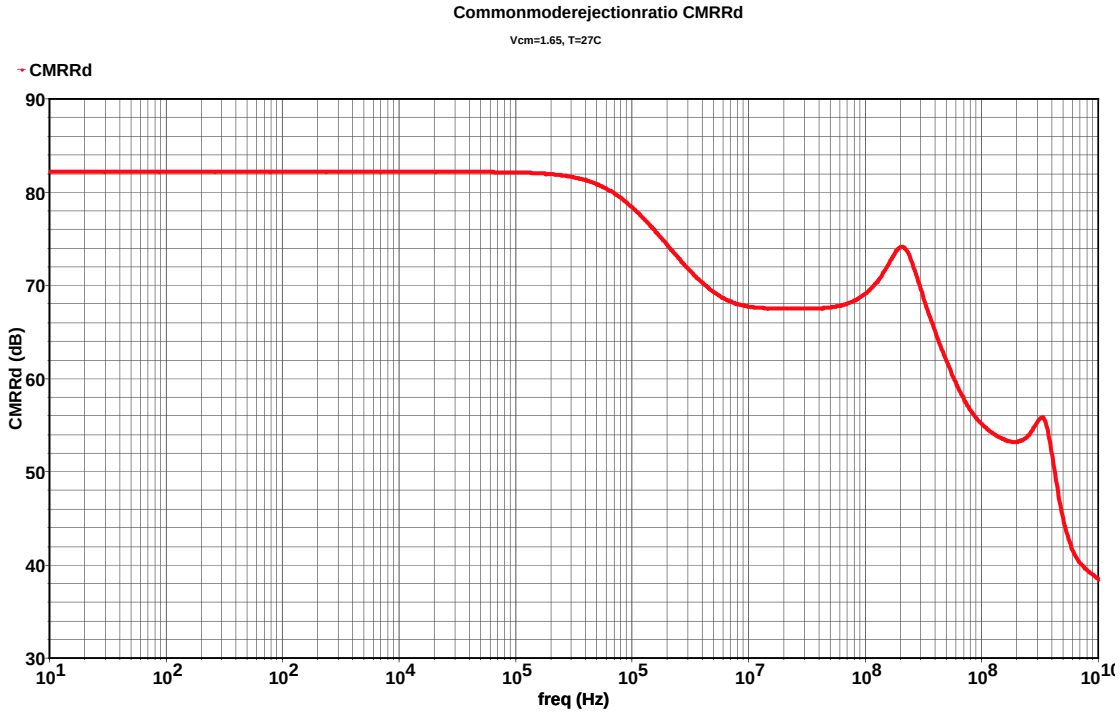



Figure 5.19: Common mode rejection ratio vs. frequency

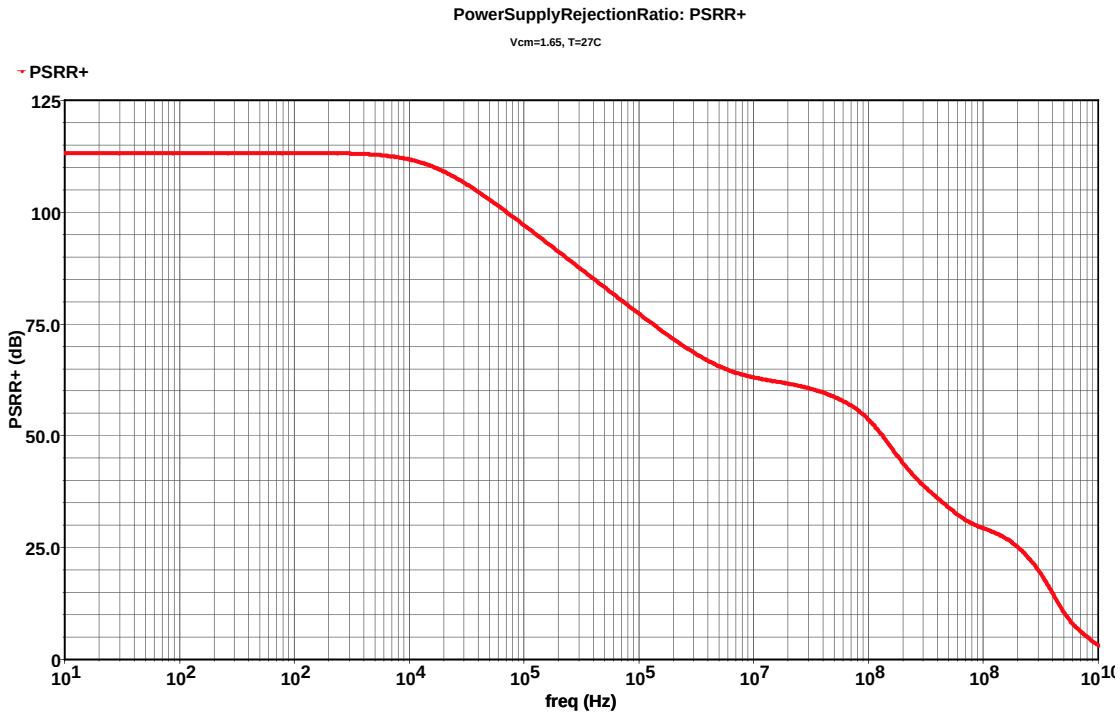



Figure 5.20: Power supply rejection ration vs. frequency

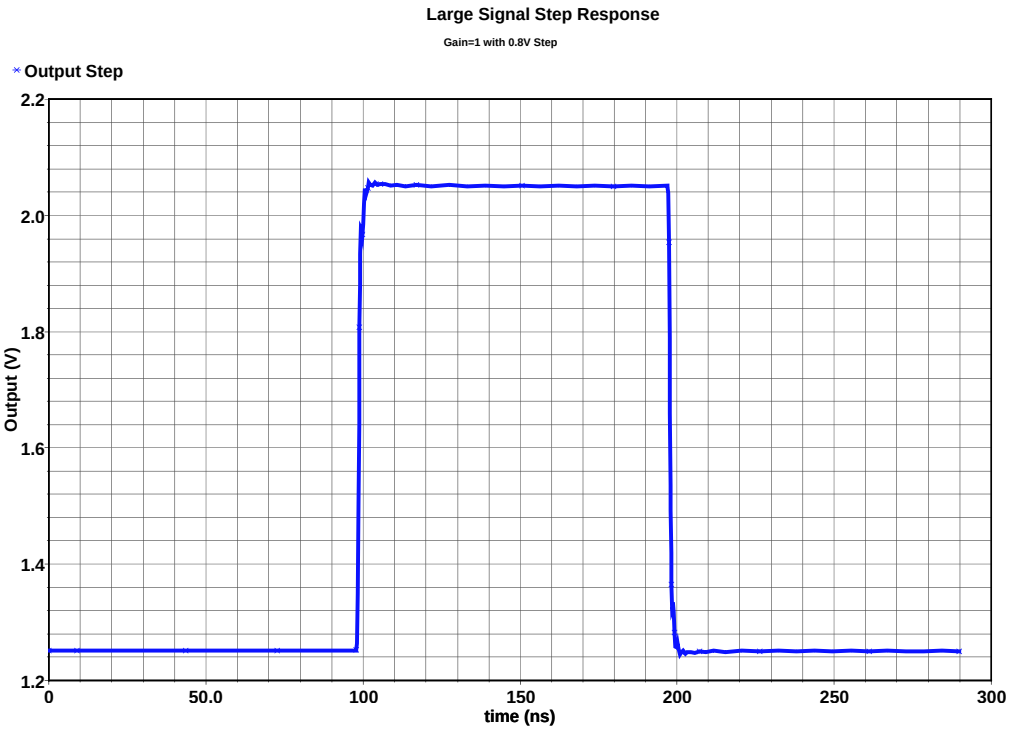



Figure 5.21: Large signal step response Gain=1 with 0.8V step

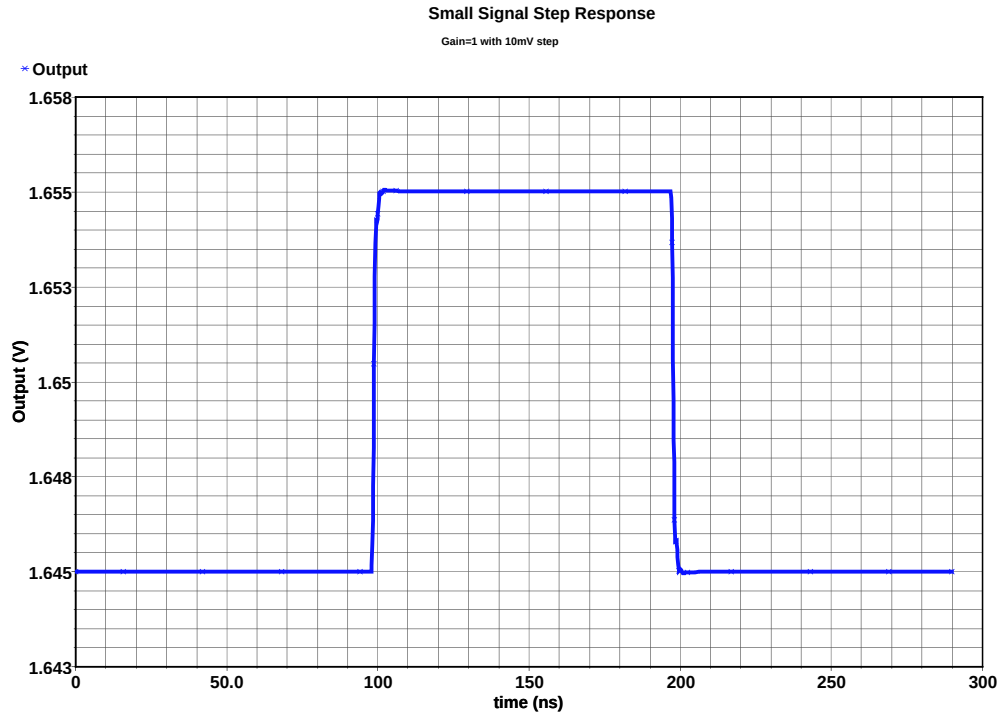



Figure 5.22: Small signal step response Gain=1 with 10mV step

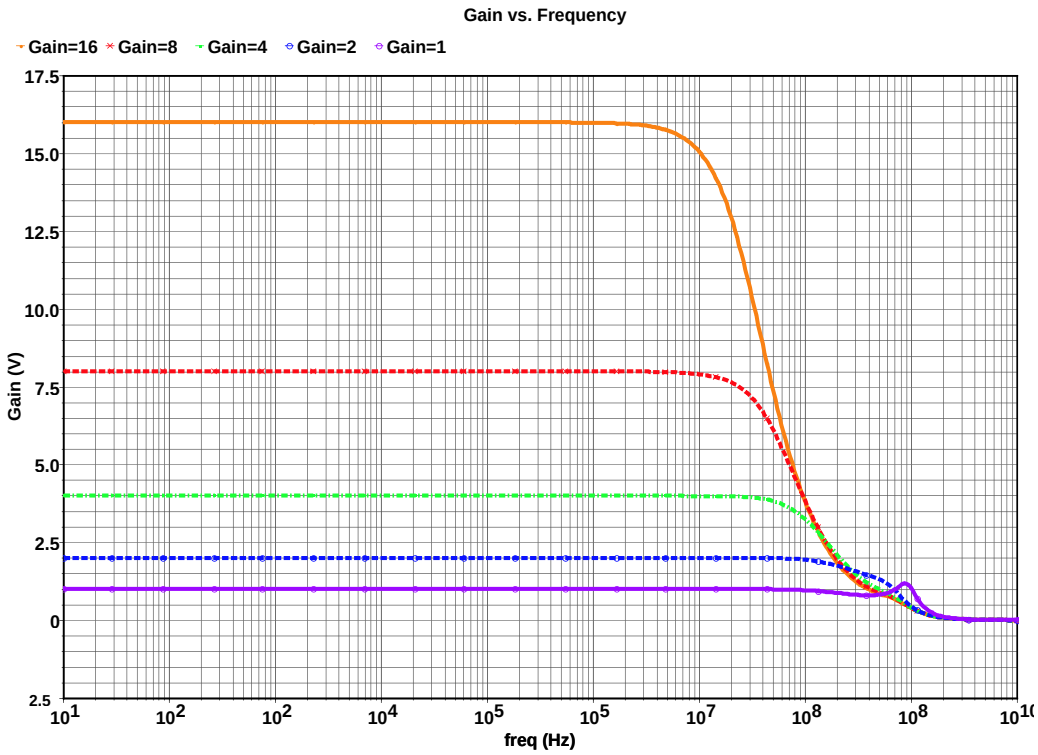



Figure 5.23: Closed-loop gain as a function of frequency

the equivalent input noise voltage is about $2.1uV$. For very low noise application, the amplifier can be further modified to improve its noise performance at the cost of area and more power consumption.

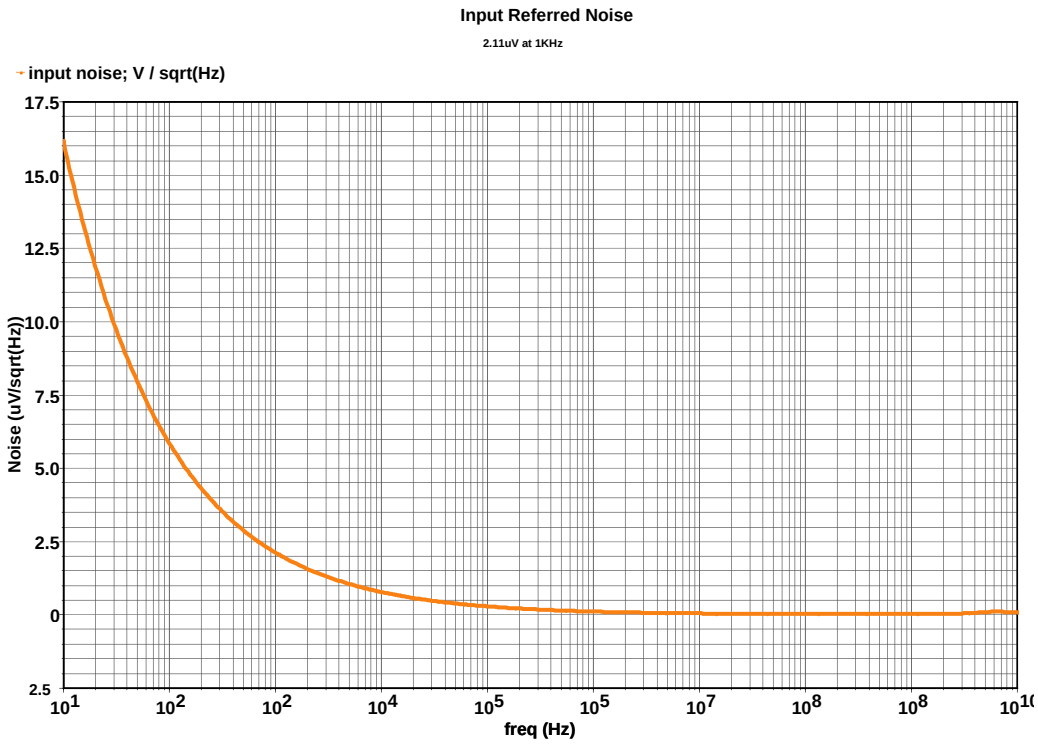



Figure 5.24: The input referred noise as a function of frequency

## 5.4 Application of Laser Makelink in the Op Amp Design

### 5.4.1 Offset Trimming

Due to the geometry mismatch and the other fabrication process variations (doping level, oxide thickness etc) induced mismatch, a relatively large input offset voltage often exists in the CMOS op amps (and comparators) comparing to that of their bipolar counterpart. The amplified input offset introduces a DC shift at the amplifier's outputs, which affects the output swing or may even drive the amplifier into nonlinear operation mode. Moreover, the input offset severely limits precision of the system. Sometimes it may impose the lower bound on the maximum resolution that the system can obtain. Thus offset cancellation technique is always employed in

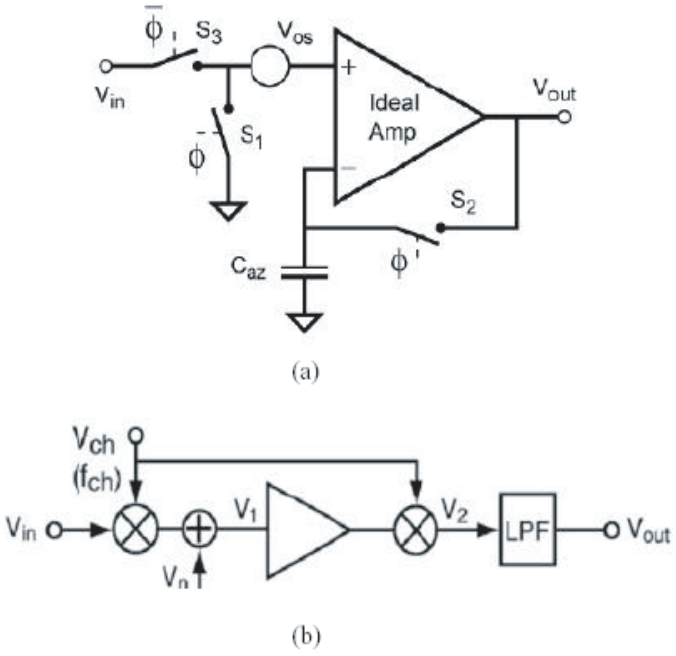



Figure 5.25: Offset cancellation (a) Auto-zeroing; (b) Chopper stabilization

high precision analog and mixed-signal designs such as analog-to-digital converters.

Traditionally offset voltage is canceled dynamically using clock controlled MOSFET switches. A periodic refreshing is required, because the junction and subthreshold leakage of switches eventually corrupts the correction voltage stored across the capacitors. In a typical design, the offset must be refreshed at a rate of at least a few kilohertz. Previous offset cancellation techniques fall into two categories: 1) autozeroing (AZ) or 2) chopper stabilization (CHS). The follow figures illustrate these two techniques [82].

AZ is essentially a sampling technique. Two clock phases are needed: (1) sampling phase: the unwanted quantity (offset and noise) is sampled and stored on the capacitors; (2) signal processing phase: this unwanted quantity is subtracted from the contaminated signal either at the input, intermediate nodes or the output

of the amplifier/comparator [83]. Unlike the AZ approach, the CHS technique does not use sampling, but rather applies modulation to transpose the signal to a higher frequency where there is no 1/f noise, and then demodulates it back to the baseband after amplification. A low pass filter is usually required to recover the desired signal.

These traditional approaches can reduce the input offset as well as the low frequency noise (mainly 1/f noise), but the drawbacks are also obvious: (a) increased circuit complexity and transistor count (b) extra clocks needed (c) more silicon real estate cost (d) increased thermal and flicker noise and power consumption (e) increased production cost. (f) degraded circuit performance (for example, reduced bandwidth).

Other offset cancellation techniques such as $Digitrim^{TM}$ from ADI and laser trimmed thin film resistor are also available. DigiTrim adjusts circuit performance by programming digitally weighted current sources [84]. The latter method is to use laser to cut the thin film resistors. As the beam traces along a resistor, it effectively changes the width of the resistor. Since the resistor's value is proportional to its width, this permanently changes the resistor's value. This requires special process steps to deposit the thin film thus increases fabrication cost. For CMOS op amp, this method is not very attractive because resistor load structure is not used very often.

On the contrary, the laser trimming technique proposed here does not have these limitations. The offset is measured and trimmed at the wafer level during production.

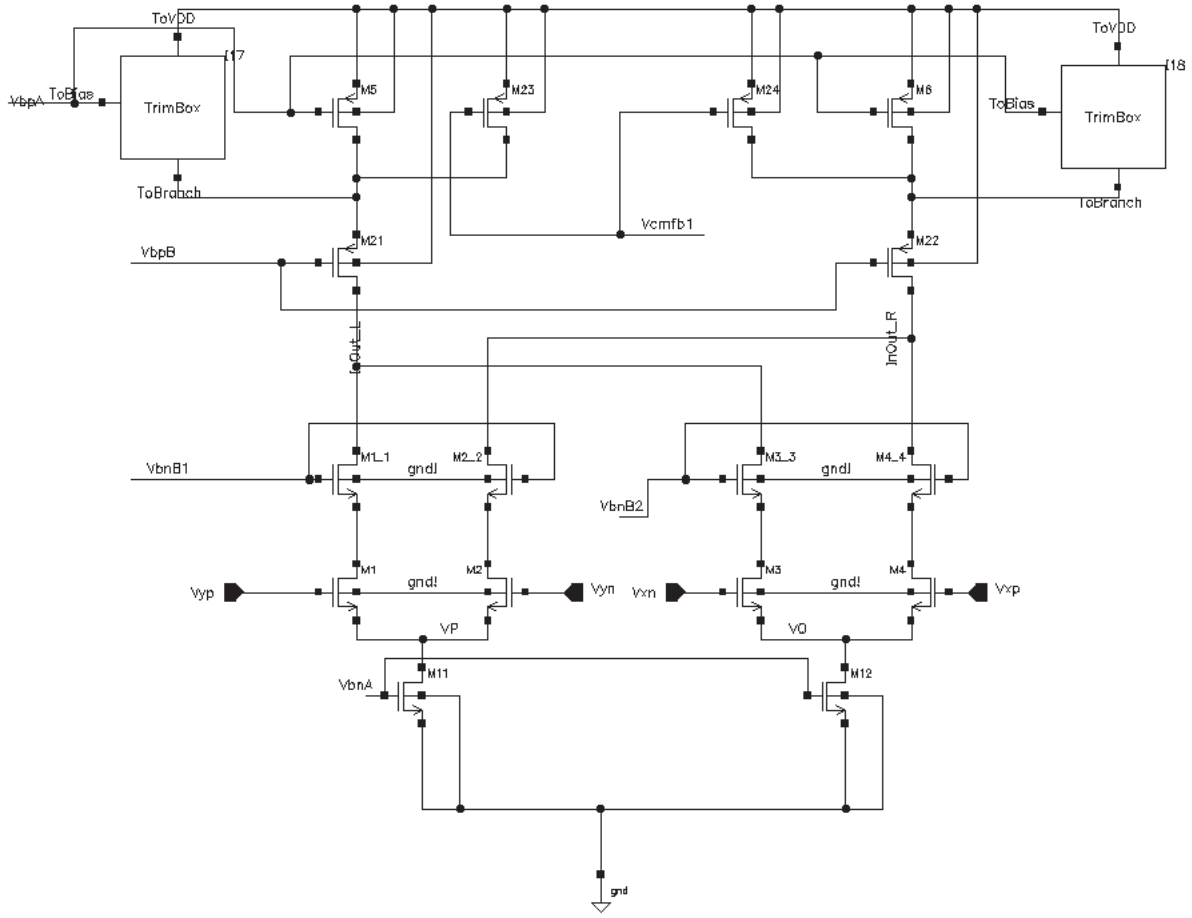
The input offset voltage is caused by process variation (doping level, litho-

graphic errors, thermal/mechanical stress etc) induced device mismatches, which include unmatched geometry sizes, threshold voltage or mobility mismatch etc. The offset voltage of the differential pair can be expressed as [75]:

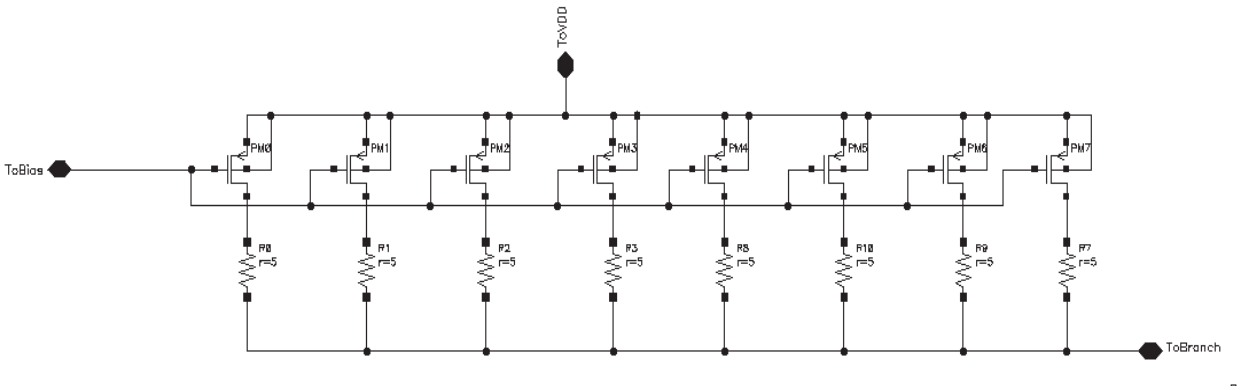
$$V_{off} = \sqrt{\frac{2I_{DS}}{\mu C'_{ox}\frac{W}{L}}}[\frac{-\Delta I_{DS}}{2I_{DS}} + \frac{\Delta(W/L)}{2(W/L)}] - \Delta V_{th} \tag{5.30}$$

$I_{DS}$ is the drain current of one branch, $\Delta I_{DS}$ is the current difference between the two branches, $\Delta V_{th}$ is the threshold voltage difference between the two input transistors, $\Delta(W/L)$ is the transistor size mismatch. Neither the first term nor the second term on the right hand side of the above equation can be controlled to zero in practical fabrication process, but their difference may be minimized so that the offset voltage value is reduced. This goal can be achieved by adjusting the current flowing in either branch. The idea is illustrated with the input stage of a fully differential op amp as shown in figure 5.26.

The $W$, $L$ or $V_{th}$ mismatch will produce unbalanced DC bias currents in the two branches. Due to the high impedance presented at nodes InOut_L and InOut_R, a fairly large voltage difference will occur at the output. Therefore, non-zero differential input voltage, i.e., the offset voltage, must be applied in order to drive the output to the desired value, which in this design is the middle of the supply rail. To compensate the input offset, some smaller size PMOS transistors were intentionally added (inside Trim Box) in parallel with the PMOS current source. These extra transistors have the same channel length (or longer channel length can be chosen to reduce the effect of loading) as the PMOS current source but binary weighted channel width, 1x $W_{min}$, 2x $W_{min}$ .... Each PMOS transistor has a laser Makelink




(a)




(b)

Figure 5.26: (a) the input stage of a fully differential CMOS op amp (b) The internal configuration of the trim box

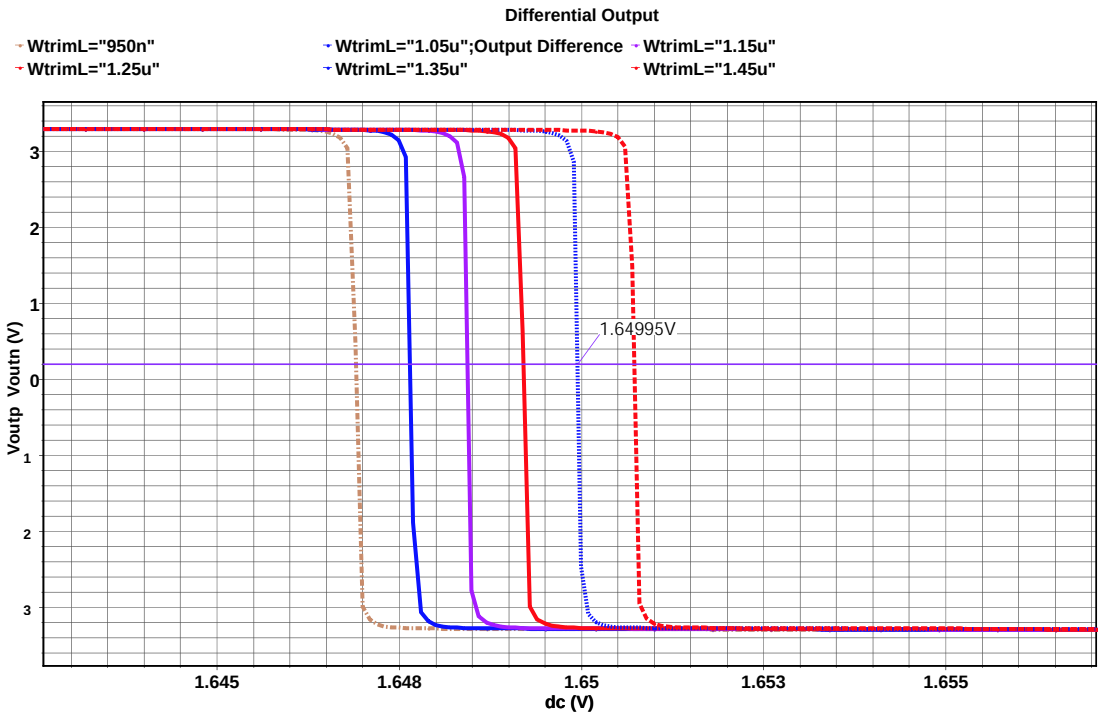



Figure 5.27: Offset trimming by laser Makelink: $10mV$ offset is reduced to $50uV$ with a group of 10 “trim” transistors

switch attached to it. One terminal of the Makelink is connected the drain, the other one is connected to the drain of the PMOS current source in the differential branch. A $5\Omega$ resistor inserted in serial with each transistor to represent the actual Makelink resistance. Before laser processing/trimming, all the switches are at off state. Once the Makelink is being zapped, the extra transistor(s) will be added to the current path. Based on the actual measurement, the number of these ”extra” transistors added to the original circuit can be controlled. This is equivalent to increasing the width of the PMOS current source at metallization level. Thus the current in the two differential branches can be adjusted with minimal effect on the circuit normal operation.

The width step of the ”extra” PMOS transistors should be determined based

on the fabrication statistics. This usually achieved by running a Monte Carlo simulation or obtained from the field data. Based on the process statistic data provided by TSMC, the maximum offset distribution was found to be below $10mV$ (1000 samples). Assuming $10mV$ offset, a group of 10 "trim transistors" with 0.1um step size is sufficient. Figure 5.28 is offset cancellation result. After laser "trimming", it can be reduced to $50uV$.

The advantages of laser trimming can be summarized as:

- Fully compatible with most of the commercial CMOS processes
- Has little or negligible effect on the amplifier/comparator's speed/bandwidth
- No aliasing or intermodulation issues
- Simple circuit scheme: a) minimum modification on the circuit topology; b) less component count thus less silicon area cost; c) less power consumption; d) very little extra thermal/flicker noise; e) no external/internal clocks needed.

### 5.4.2 Laser Reconfiguration

The application of laser Makelink is not just limited to "trimming". The flexibility of this technology can be seen by its capability to reconfigure the circuit topology at "equivalent-to-mask" level after fabrication.

For an on-chip, internally compensated op amp, to ensure the circuit stability, it's often designed to be over-compensated. The side effect of this over-compensation is it sacrifices op amp's bandwidth and speed for stability. Or, if the design under-

estimates the loading capacitance and does not provide enough phase margin, then the op amp will be unstable. To remedy this over- or under-compensation, multiple compensation capacitors may be placed in parallel on the path, as shown in figure 5.28. In this example, three capacitors with value of $100fF$, $150F$ and $200fF$ are used in the compensation capacitor bank. According to the application specific loading capacitance, the achievable bandwidth can be optimized. The default compensation capacitance value is $250fF$. If in some case, a very large $C_L$ is present at the output node, for example, $C_L = 2pF$. The $250fF$ compensation only provides $42.7^\circ$ phase margin, which may not be sufficient. To gain more phase margin, a $450fF$ compensation can be obtained by connecting three capacitors in parallel. The achieveable phase margin is improved to $64.8^\circ$.

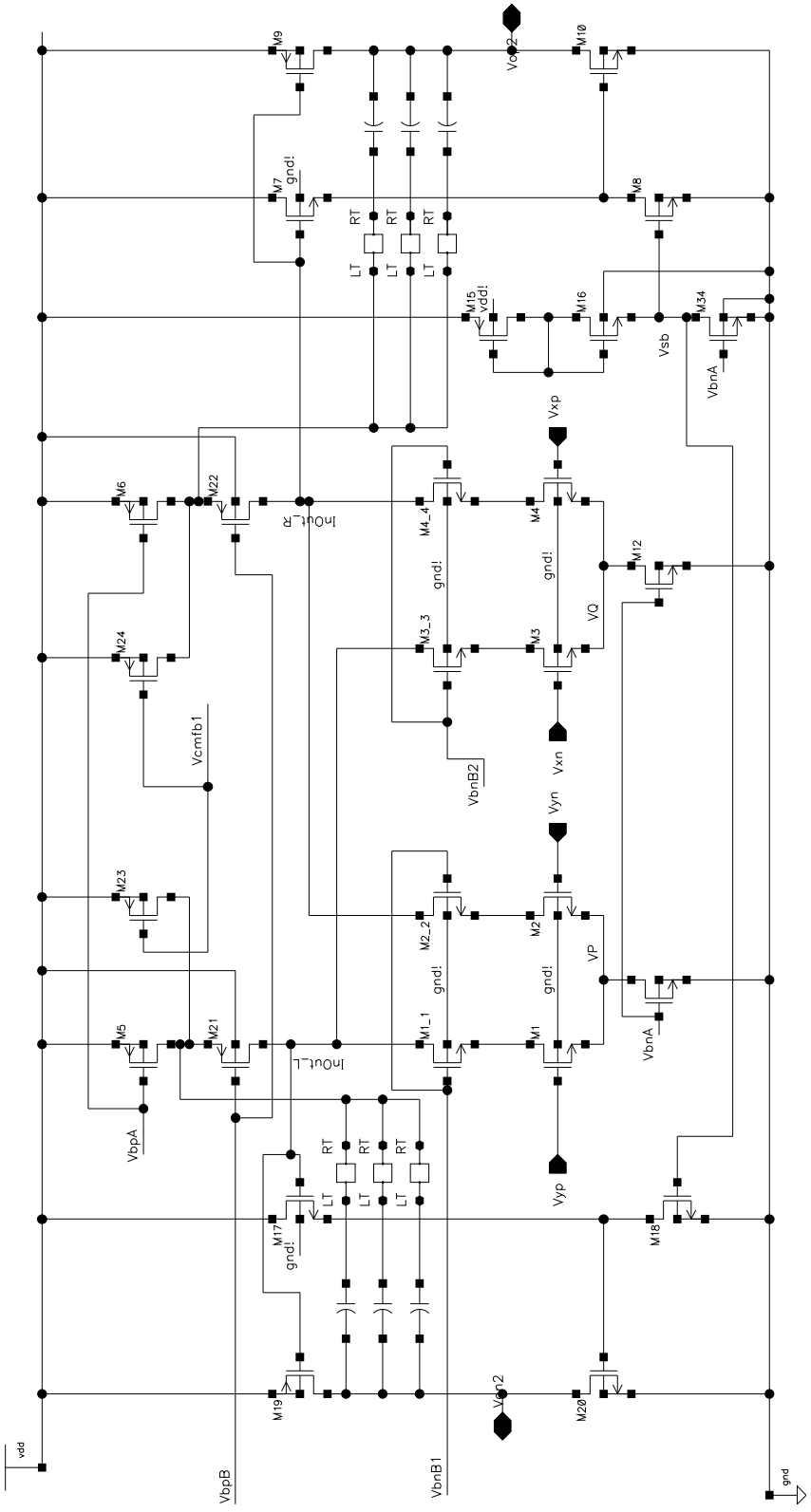



Figure 5.28: The amplifier core showing multiple compensation

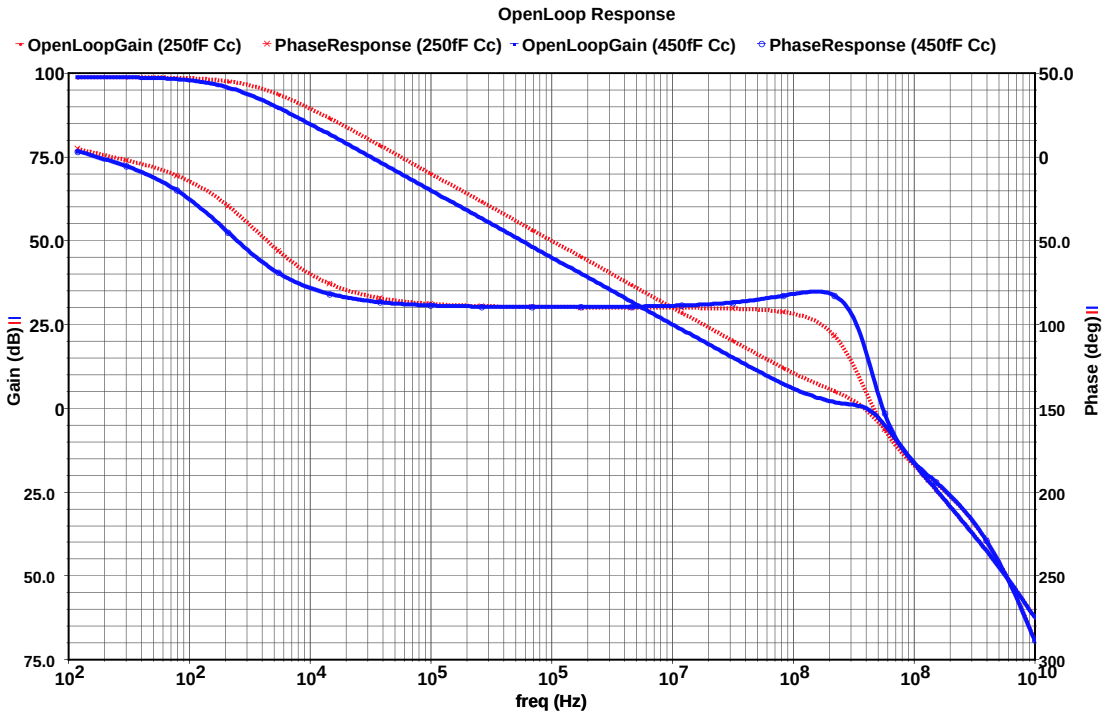



Figure 5.29: DDA open-loop frequency response: 250fF Cc vs. 450fF Cc

# Chapter 6

# Bandgap Reference

## 6.1 Introduction

Voltage and current references are pivotal building blocks in Analog/Mixed-Signal/RF designs. Not only are they used as stand-alone ICs, but they are also used within the designs of many other ICs (Figure 1). They exist in the power management block, data converter reference and amplifier biasing network. Sometimes they may have a major impact on the performance and accuracy of the whole system. For example, a voltage reference is often required in high resolution data converters to quantify the input signal. A ±1.2mV tolerance on a 1.2V reference corresponds to ±0.1% accuracy. This limits the resolution to approximately 10 bits. The bandgap references developed here can be used as common-mode reference for the DDA, to generate data converter reference voltage, or it may be combined with other components in the FPAA to implement various applications.

In general such reference circuits generate an DC quantity, which exhibits little dependence on process parameters, supply voltage or temperature (PVT)[1].

[1]Some references have a well-defined dependence on the temperature instead of temperature independence, for example, a quantity that is directly proportional to absolute temperature (PTAT). This kind of circuits are widely used in the temperature monitoring systems.

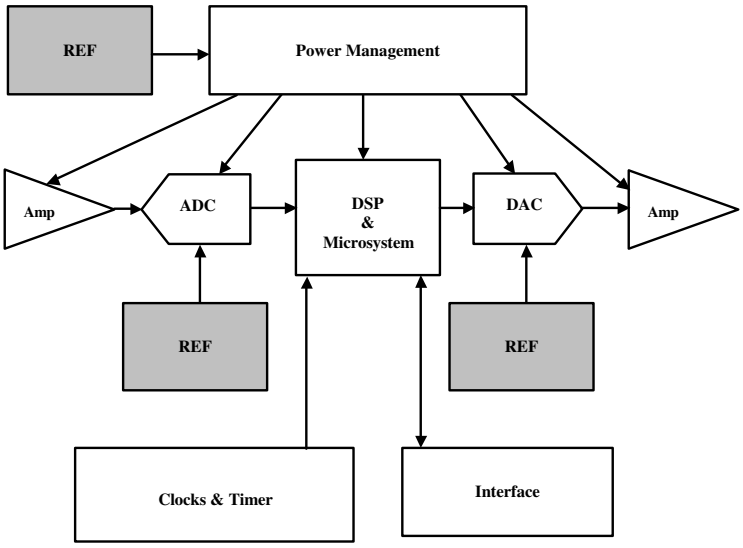



Figure 6.1: A generic Mixed-Signal System

There are many reference topologies available based on the different applications and process technologies. Figure 2 shows two simple implementations.

Although it's somewhat decoupled from the power supply rail, Figure 2 (a) still has many deficiencies as a reference. The $V_{BE}$ value highly depends on process parameters and has a very strong temperature coefficient (TC) of about -3.3%/°C. Figure 2 (b) shows a better implementation. A Zener [2] diode is used in conjunction with a regular diode, and an appreciably higher output voltage is realized. Since Zener diode has a positive TC between +1.5 and +5 mV/°C, the overall TC of the reference can be reduced to 100ppm/ °C or less with proper bias and care-

[2]Pure Zener breakdown usually occurs below 5 V. Nowadays diodes with well-defined breakdown characteristics are all called Zener diodes even though their breakdown mechanism is actually avalanche breakdown. They typically have a breakdown voltage between 5 V and 8.5 V.

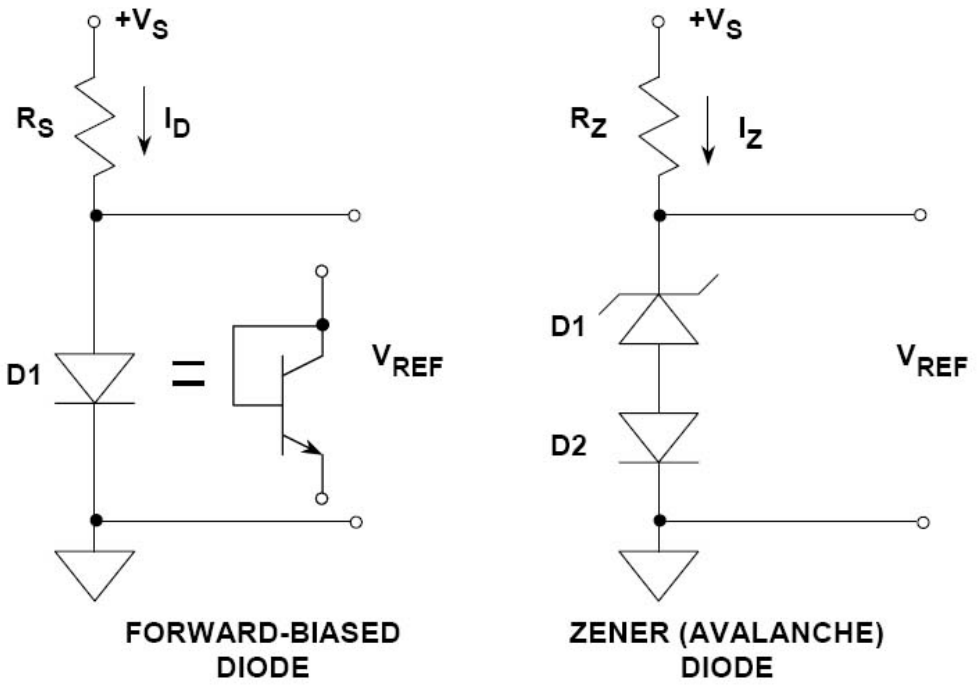



Figure 6.2: Diode References

fully chosen diodes. However, Zener diode based references must be driven from a supply voltage considerably higher than 5 V, and it's not fully compatible with modern CMOS process. Also, Zener diodes are noisy due to the noisy nature of the (avalanche) breakdown mechanism. Therefore they are not adequate for high performance CMOS ICs.

## 6.2 Principle of Bandgap Reference

The bandgap reference (BGR) circuit has been the most elegant way to implement a stable, accurate and temperature independent low voltage reference on silicon. The principle relies on the proper summation of a complementary-to-absolute-temperature

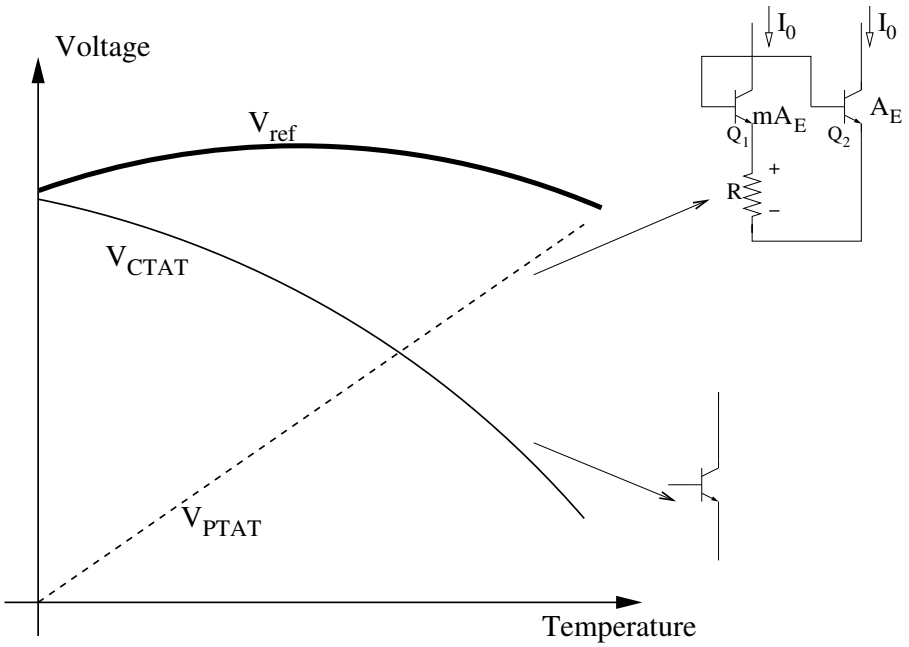



Figure 6.3: An Illustration of Bandgap Principle

(CTAT) voltage with a proportional-to-absolute-temperature (PTAT) voltage, as shown in Figure 3 and equation (3.1):

$$V_{ref} = V_{PTAT} + V_{CTAT} = V_{BE} + x\Delta V_{BE} \tag{6.1}$$

so that $\partial V_{ref}/\partial T = 0$. Here $x$ is a weighting factor.

The PTAT voltage can be generated by two BJT's operated at different current densities:

$$V_{BE1} = V_T \ln (J_1/J_s) \tag{6.2}$$

$$V_{BE2} = V_T \ln (J_2/J_s)$$

Then:

$$\Delta V_{BE} = V_T \ln (J_1/J_2) = V_T \ln m \tag{6.3}$$

and:

$$\frac{\partial \Delta V_{BE}}{\partial T} = \frac{k}{q} \ln m \tag{6.4}$$

where $k$ is Boltzmann constant, $q$ is electron charge, $J_C$ is current density and $V_T$ is thermal voltage. This shows $\Delta V_{BE}$ is PTAT. Note here $J_C = \Delta V_{BE}/(RA_E) = V_T/(RA_E)$ is also an PTAT quantity, if neglecting the resistor $R$'s temperature dependence for the moment.

It is well known that $V_{BE}$ has a negative TC with nonlinear temperature dependence. This can be found by taking derivative of equation (2) (assuming linear temperature dependence of $J_C$):

$$\begin{aligned} \frac{\partial V_{BE}}{\partial T} &= \frac{\partial V_T}{\partial T} \ln \frac{J_C}{J_s} + \frac{V_T}{J_C}\frac{\partial J_c}{\partial T} - \frac{V_T}{J_S}\frac{\partial J_S}{\partial T} \\ &= \frac{k}{q} \ln \frac{J_C}{J_s} + \frac{k}{q} - \frac{V_T}{J_S}\frac{\partial J_S}{\partial T} \end{aligned} \tag{6.5}$$

The first two terms in equation (5) represent the part of linear temperature dependent behavior of $V_{BE}$, while the 3nd term represents the higher order temperature dependence. The saturation current $J_S$ can be approximated by [85]:

$$J_S \approx C_1 T^{\eta} \exp\left(\frac{-E_G(T)}{kT}\right) \tag{6.6}$$

where $\eta$ is a process-dependent parameter (representing the temperature dependence of intrinsic carrier concentration $n_i$ and mobility $\mu$), $E_G(T)$ is silicon bandgap at temperature $T$. Substitute equation (6) into (5) and re-arrange these three terms:

$$\frac{\partial V_{BE}}{\partial T} = \frac{V_{BE}(T) - (\eta - 1) - V_G(T)}{T} \tag{6.7}$$

From equation (7), it's obvious $V_{BE}$ has a non-linear temperature dependence. But to the first order, the variation of $V_{BE}$ with temperature can be approximated as

linear with TC between 1.5 $mV/°C$ and 2.2 $mV/°C$, depending on the process parameters and temperature. Using equation (1) and by properly choosing a weighting factor $x$ (typical value is $\sim 23$), a reference with near zero TC can be obtained.[3]

The bandgap reference technique is attractive in IC designs because of its simplicity, the avoidance of Zener diodes and their noise, and more importantly these days low voltage operation (<5 V). In 1971, Widlar introduced the first bandgap reference, LM113 [86]. It used conventional junction-isolated bipolar IC technology to make a stable 1.220V reference. However most of today's bandgap references are based on the classical topology invented by Brokaw in 1974 [87] as shown in Figure 4. The two BJT's Q1 and Q2 with different emitter area (1:8) are operated at same collector current by virtue of the closed loop around the buffer amplifier and $R_8 = R_7$, thus here m is 8 (equation (2)). The PTAT voltage $\Delta V_{BE}$ drops on resister $R_2$ and defines the current $I_2 = I_1 = \Delta V_{BE}/R_2$. The voltage drop across resistor $R_1$ is:

$$V_1 = 2\frac{R_1}{R_2}\Delta V_{BE} \tag{6.8}$$

The resistors may have very high TC, but their ratio should be nearly temperature independent. So $V_1$ is PTAT. The bandgap voltage $V_2$ is determined by:

$$\begin{aligned} V_Z &= V_{BE1} + V_1 \\ &= V_{BE1} + 2\frac{R_1}{R_2}V_T \ln 8 \\ &= 1.205V \end{aligned} \tag{6.9}$$

[3]Since most process parameters vary with temperature, if a quantity is temperature independent, it's usually also process independent.

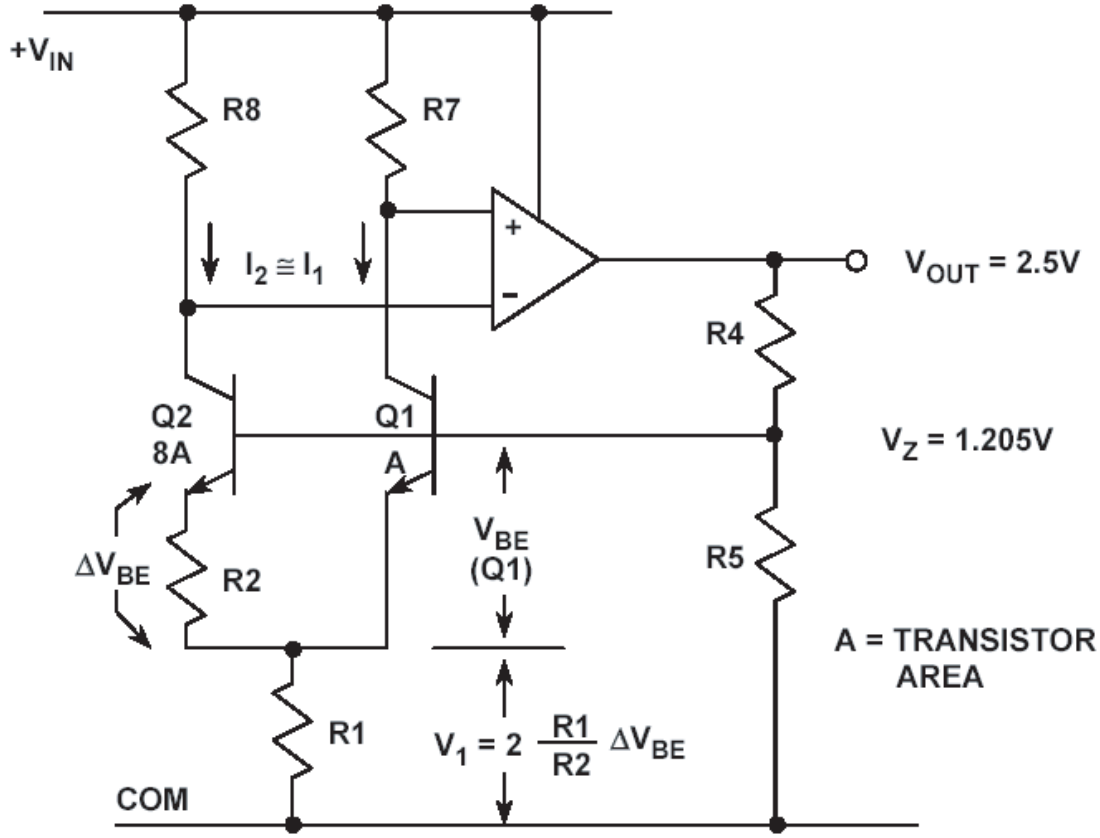



Figure 6.4: AD580 Precision Bandgap Reference Based on Brokaw Cell, Analog Devices, 1974

The output voltage can be scaled up using the buffer amplifier and the resistor ladder. This is a first order bandgap reference because it only compensates the linear component in equation (7).

## 6.3 A CMOS Implementation of Bandgap Reference

The goal of this work is to develop an CMOS bandgap reference suitable for Field Programmable Analog Array and its associated applications. Apparently high

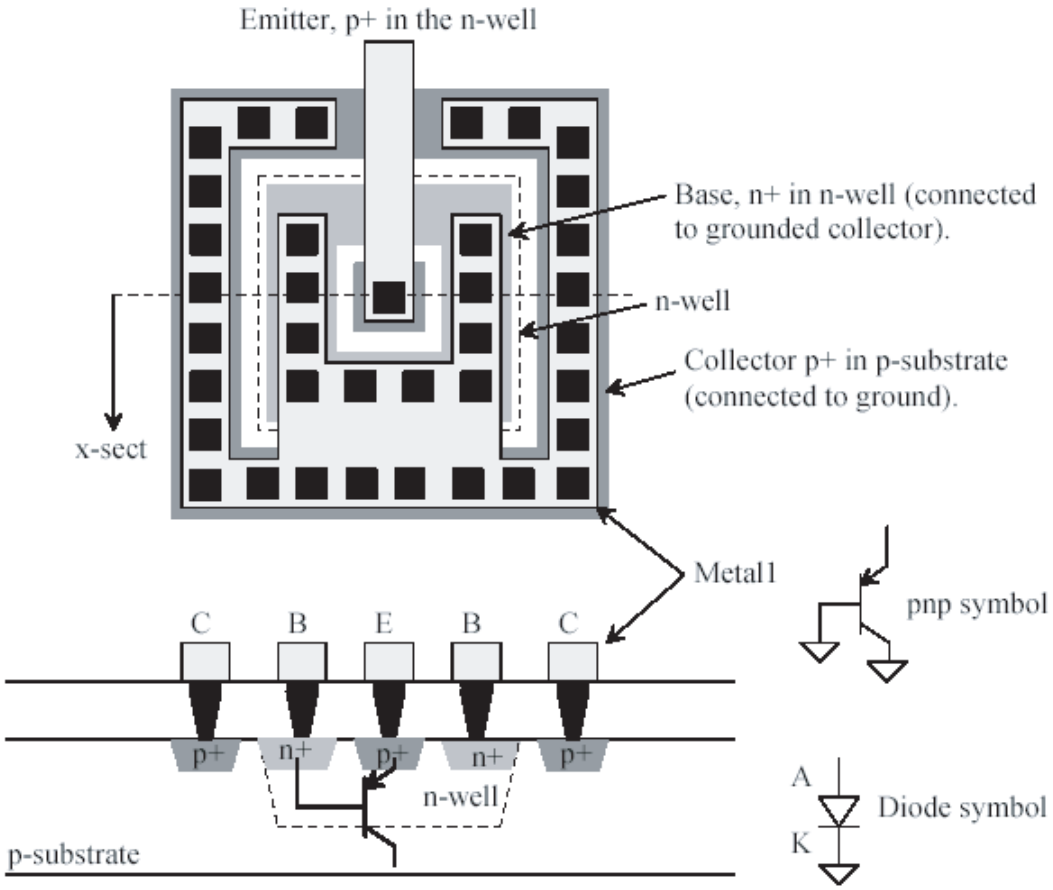



Figure 6.5: Realization of Substrate PNP BJTs on the CMOS process [88]

performance BJT's are not available on the standard CMOS processes. Therefore the classical Brokaw cell cannot be implemented directly in the original form. Fortunately, for CMOS bandgaps, the parasitic substrate PNP transistors (Figure 5) can be used. Even though they have a low $\beta$, "a poorly performing bandgap reference is still considerably superior to anything that can be built out of pure CMOS components".

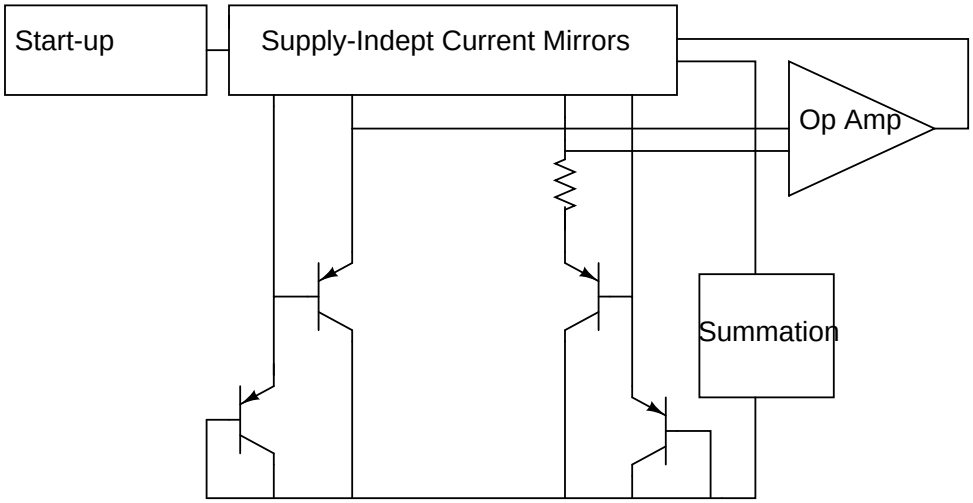



Figure 6.6: A Block Diagram of the Proposed BGR

### 6.3.1 Architecture

A block diagram of the proposed BGR is shown in Figure 6. It consists of a bandgap core ($Q0$ through $Q3$) which generates the PTAT and CTAT voltages; a high-gain op amp which is used to maintain nodes $A$ and $B$ at same potential and regulate the current mirror biasing point to suppress supply voltage variation; a summation branch generating the final bandgap voltage; and at last a start-up circuit to ensure the BGR operates properly at power-up. It should be noted that instead of one $PN$ junction there are two $V_{BE}$'s stacked together in each of the branches. Besides it can directly provide a higher output reference voltage ($\approx$2.5V, about twice the value of the general structure), an added advantage is this topology can lower the effect of the op amp offset error.

### 6.3.2 The Bandgap Core

Figure 7 shows the BGR core circuit. It contains two pairs of stacked diode-connected substrate *pnp*'s. Transistors $Q1$ and $Q3$ in conjunction with $Q0$ and $Q2$ are used to develop the PTAT voltage. The emitter area of the four transistors was set as: $A_{E1} = A_{E3} = 4A_{E0} = 4A_{E2}$. Using TSMC018 CM process model parameters, the TC of $V_{BE}$ was found to be $\approx$1.80 $mV$ at room temperature. The four identical PMOS transistors have channel length of 4 $um$ to reduce channel length modulation effect. An 40 $uA$ biasing current with a relatively high overdrive voltage ($V_{ov} \approx 0.55$ $V$) was picked to improve the matching between the current mirrors. Since each of the branches carries the same current and nodes $A$ and $B$ have the same potential due to the negative feedback around the op amp, the two $V_{BE}$ difference drops across resistor $R_1$:

$$V_{R1} = (V_{BE1} - V_{BE0}) + (V_{BE3} - V_{BE2}) = 2V_T \ln 4 \tag{6.10}$$

This defines the PTAT voltage. The drain-source voltages of the PMOS transistors are matched well so the current systematic offset won't be an issue. As will be explained later, the accuracy and temperature dependence of the resistor $R_1$ are not a problem. The primary error source here is the op amp offset, which causes a finite potential difference between nodes $A$ and $B$:

$$V_{ERR} = V_{OFF} + \frac{V_C}{A_m} \tag{6.11}$$

where $V_{OFF}$ is the op amp offset voltage, $A_m$ is the gain. This error voltage $V_{ERR}$ should be kept small comparing to the PTAT voltage $2V_T \ln(4)$. This explains why two stacked PN junctions can lower the effect of op amp offset error.

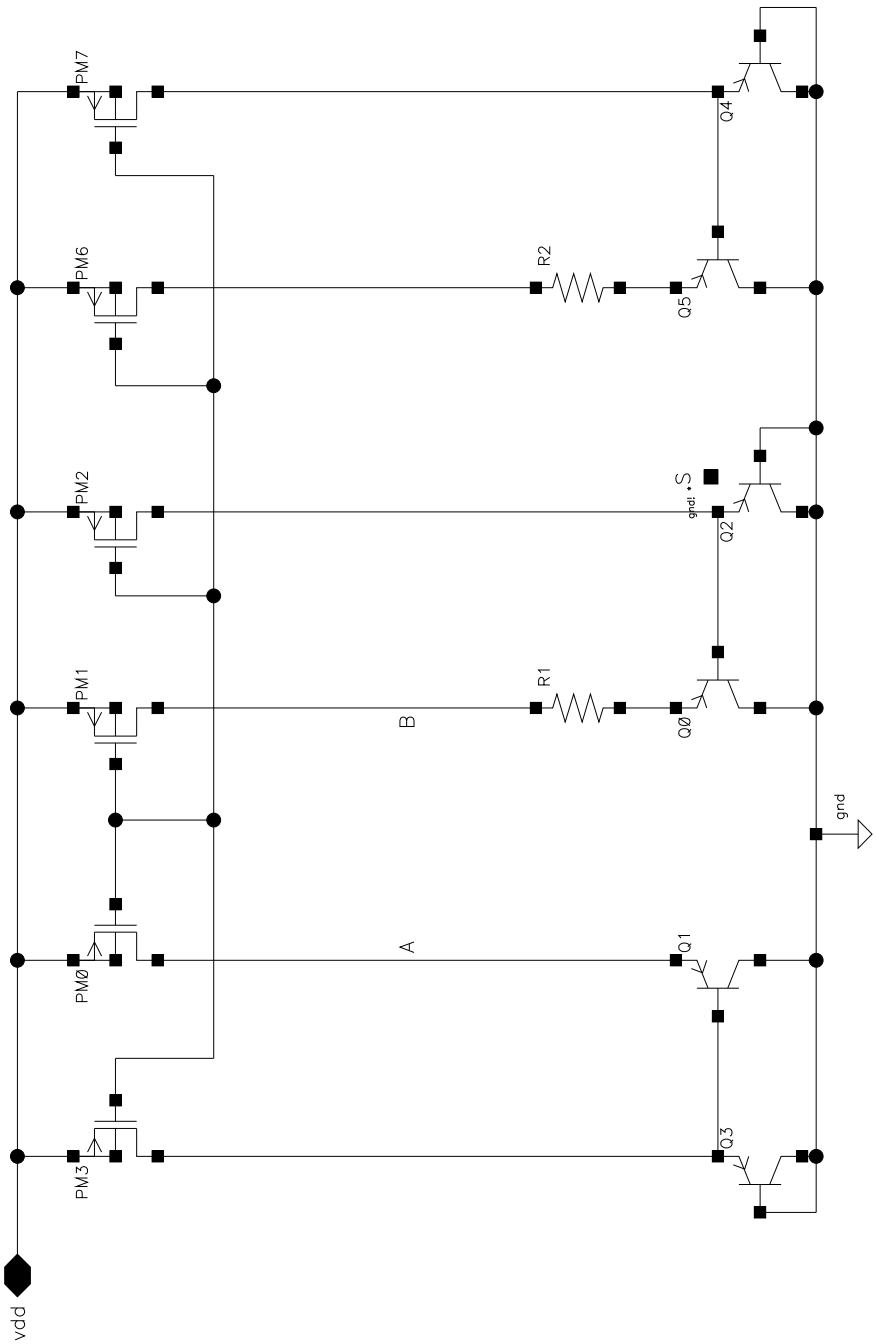



Figure 6.7: The BGR Core

### 6.3.3 Op Amp Design

The op amp plays an important role in the BGR. By intuition a high-gain, low-offset op amp would be desired. High-gain can be achieved through cascoding or a two-stage structure. For op amp offset, large transistor size for the input differential pair may be used. With judicious layout design such as interdigitation and common-centroid structure, offset can be reduced. Alternatively, autozeroing or chopper stabilization techniques can be employed [89], [90]. But seems these two methods are not widely used in BGR due to the cost of complexity, more power consumption and switching noise. As discussed in the previous chapter, Laser Makelink based trimming approach would be an excellent choice. It can also be used to trim the poly-resistor to get high precision bandgap voltage. Another factor that needs to be examined is the input common-mode range (ICMR). By inspecting Figure 7, ICMR is found to be within $2V_{BE} < ICMR < VDD - V_{Dsat}$. Other concerns including noise and power consumption are especially important for low voltage operation.

With above considerations, a folded-cascode two-stage topology was adopted. Two NMOS transistors are used as the input differential pair for their speed and large $g_m$. Their common-mode operation voltage falls well within the required ICMR. The amplifier has a very high gain ($\approx$ 133dB at DC), at the mean time, provides enough bandwidth ($\approx$ 65 MHz unity-gain bandwidth with 2 $pF$). An effort was also made to reduce the $\frac{1}{f}$ noise in the first stage by choosing fairly large PMOS transistors and overdrive voltage. Comparing to telescopic structure, the folded-cascode can save one $V_{dsat}$ drop. Thus it can be easily modified for TSMC 018 CM process 1.8

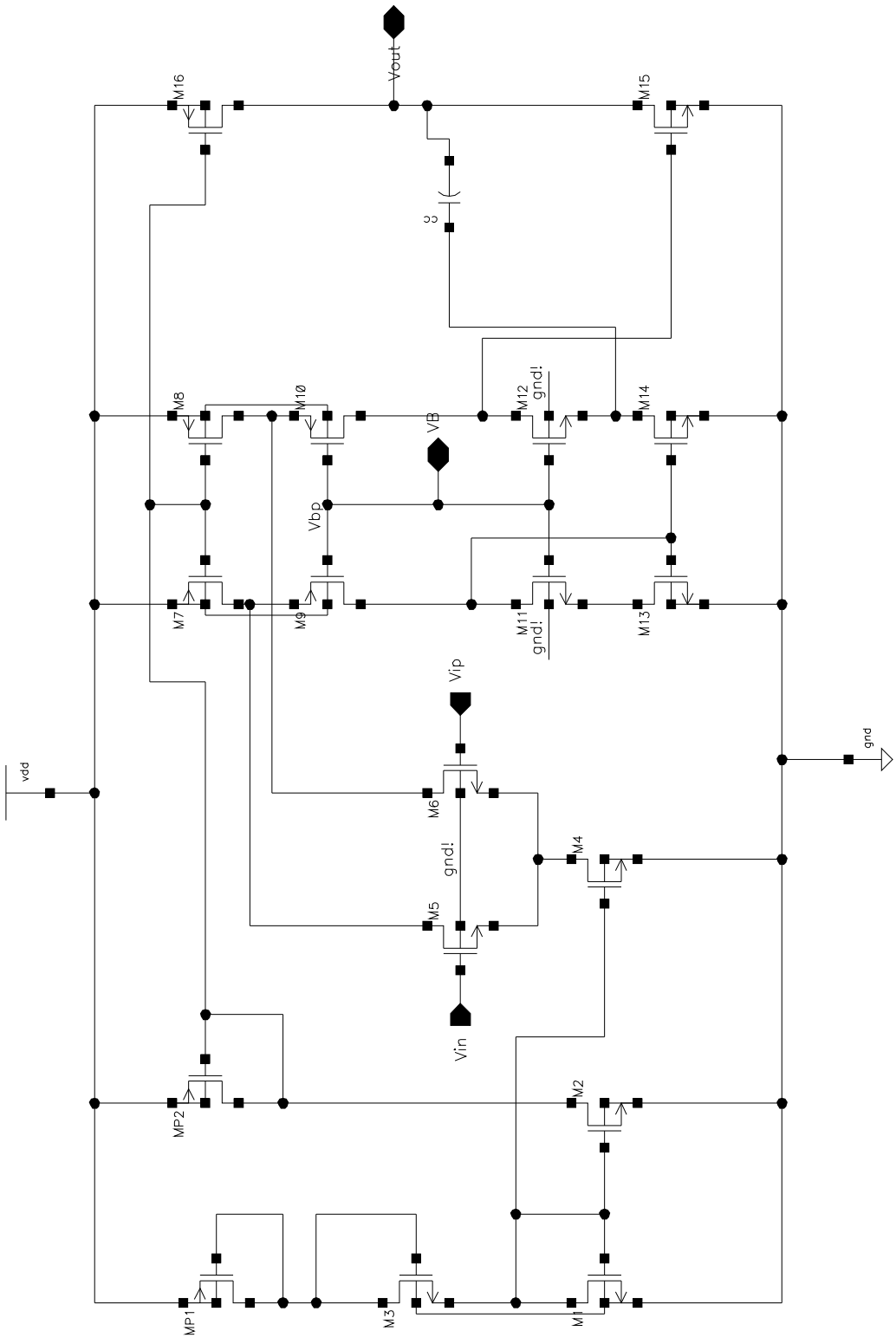



Figure 6.8: Schematic of the 2-stage folded-cascode op amp

$V$ or sub-1.5 $V$ operation. For low to medium precision system, a simple one stage error amplifier could be used. However, for stability reason, a fairly large capacitor has to be inserted between the gates of the PMOS mirrors and VDD (or gnd) to make a dominant pole compensation. This not only occupies more area but also makes the circuit susceptible to the coupling noise from the power lines through the large compensation capacitor.

The op amp schematic is shown in Figure 8. Figure 9 and figure 10 are the frequency response of this amplifier with 2 $pF$ and 10 $pF$ capacitive load, respectively. They show DC gain of 133 $dB$, unity-gain frequency of 65 MHz with 83 ° phase margin when 2 $pF$ loading capacitance is present. Even with 10 $pF$ load, this amplifier still has enough phase margin of 60°.

### 6.3.4 The Complete Circuit

The complete BGR circuit is shown in figure 11. The branch consisting BJT Q4 and Q5 ($A_{E4} = A_{E5} = 4A_{E1}$) and PMOS transistors PM6 and PM7 is the summation block. The special arrangement of resistor $R_1$ and $R_2$ will be explained in the layout section. The final BGR output voltage is defined by:

$$V_{ref} = V_{BE4} + V_{BE5} + I_{DS6}R_2 \quad = \quad 2V_{BE} + \frac{R_2}{R1}2V_T \ln m \tag{6.12}$$

Ideal resistors should have low voltage and low temperature coefficients. In this design, N+ poly resistors without silicide were used. They can be fabricated with better accuracy comparing to the N-Well resistors (±15% vs ±22.7%). Also, they have a reasonable sheet resistance of 292Ω/□. For the resistor values in the BGR,

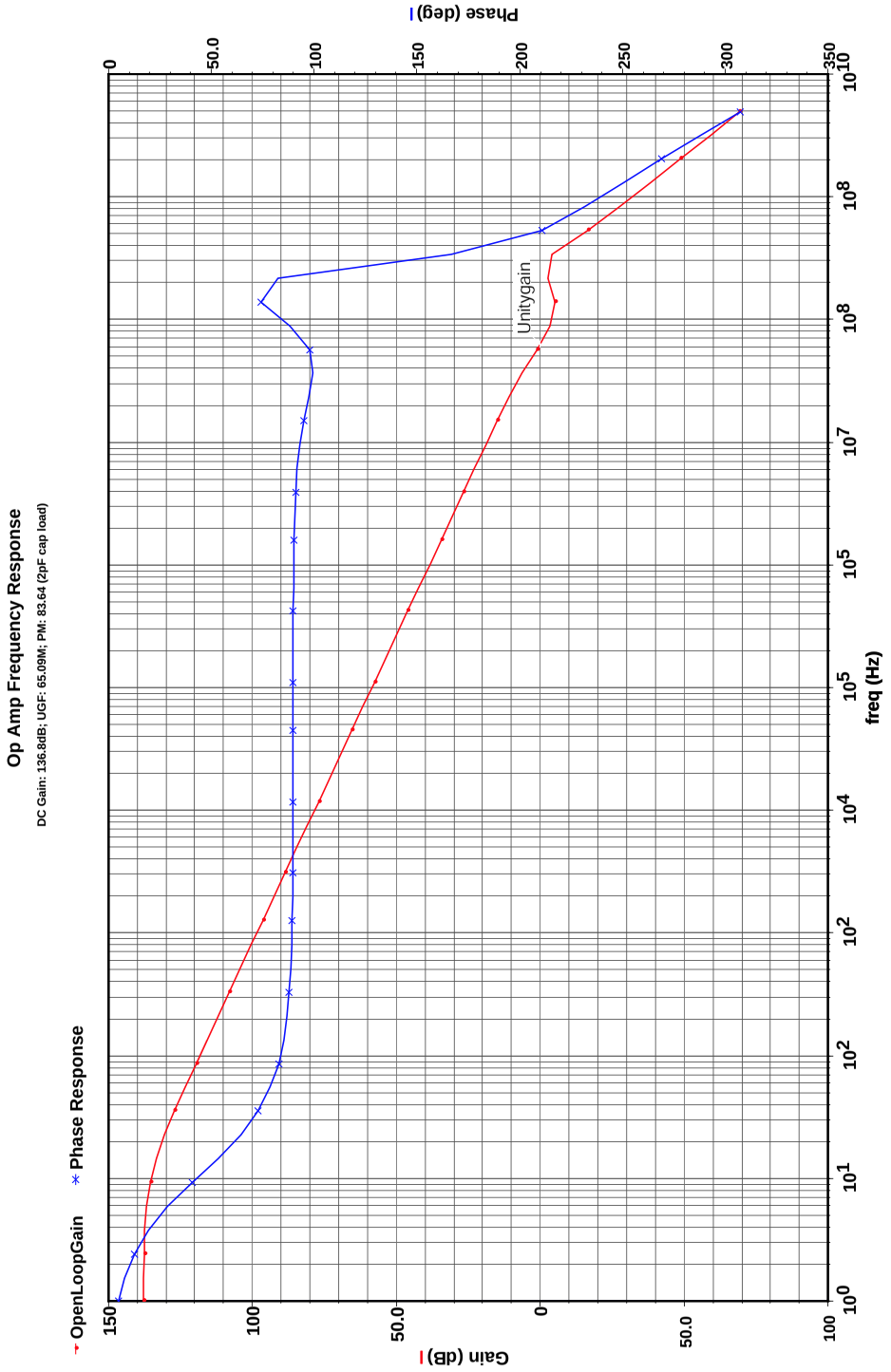



Figure 6.9: Op Amp frequency response with 2pF capacitive

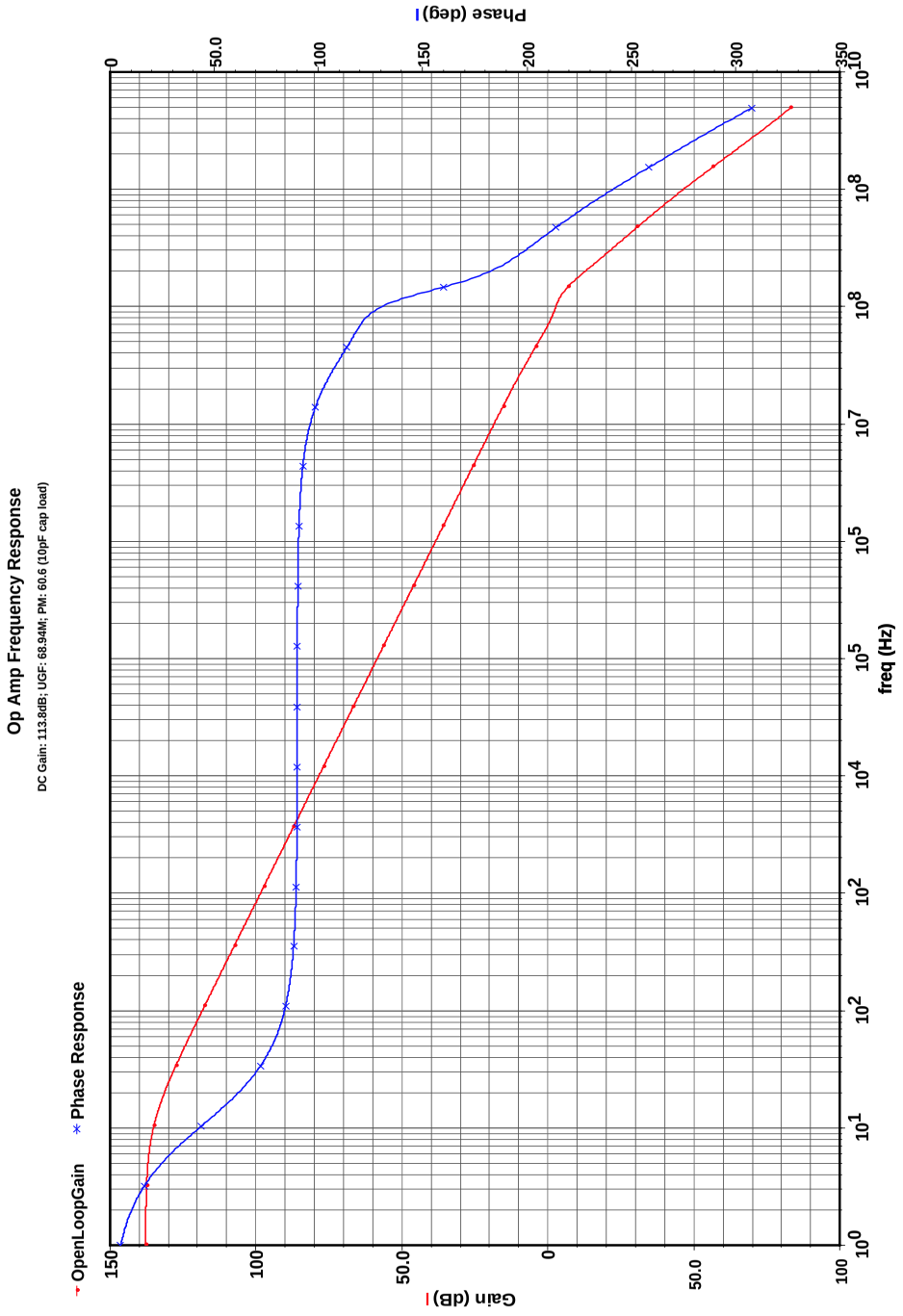



Figure 6.10: Op Amp frequency response with 10pF capacitive load

this ensures the resistor layout spanning long enough for good accuracy without taking too much space. The non-silicide poly resistors can be modeled as:

$$R = R_0[1 + VC_1\Delta V + VC_2(\Delta V)^2][1 + TC_1\Delta T + TC_2(\Delta T)^2] \tag{6.13}$$

where $\Delta T = T - 25°$.

$$R_0 = R_\square(L - \Delta L)(W - \Delta W)$$

VC, TC are voltage coefficient and temperature coefficient, respectively. $R_\square$ is sheet resistance, $\Delta L$, $\Delta W$ are length and width variations, and $R_0$ is the nominal layout dependent resistance. Once the BGR is in normal operation, the voltage variation across the resistors will be very small. So the VC is negligible. Because the resistors are made of the same material, their ratio should be nearly temperature dependent. The main error source here is the geometry/process variation caused mismatch. This kind of mismatch can be minimized with careful layout technique. By fine tuning the resistor ratio, an accurate, temperature insensitive reference voltage can be generated.

Although BGR is essentially DC-operated, there are two important dynamic issues related to the proper operation of BGR circuits: their behavior at start-up, and their behavior in response to the transient loads. For example, when the gate voltages of the PMOS current mirrors are VDD and the their source voltages are 0 (ground), there will be no current flowing through all the branches. This is a possible and stable operating point. Thus, like most of the self-biasing or bootstrap topologies, a start-up circuit is necessary to ensure the normal operation of BGR. Transistors $ST_1$ through $ST_4$ perform this function. Initially all the transistors are

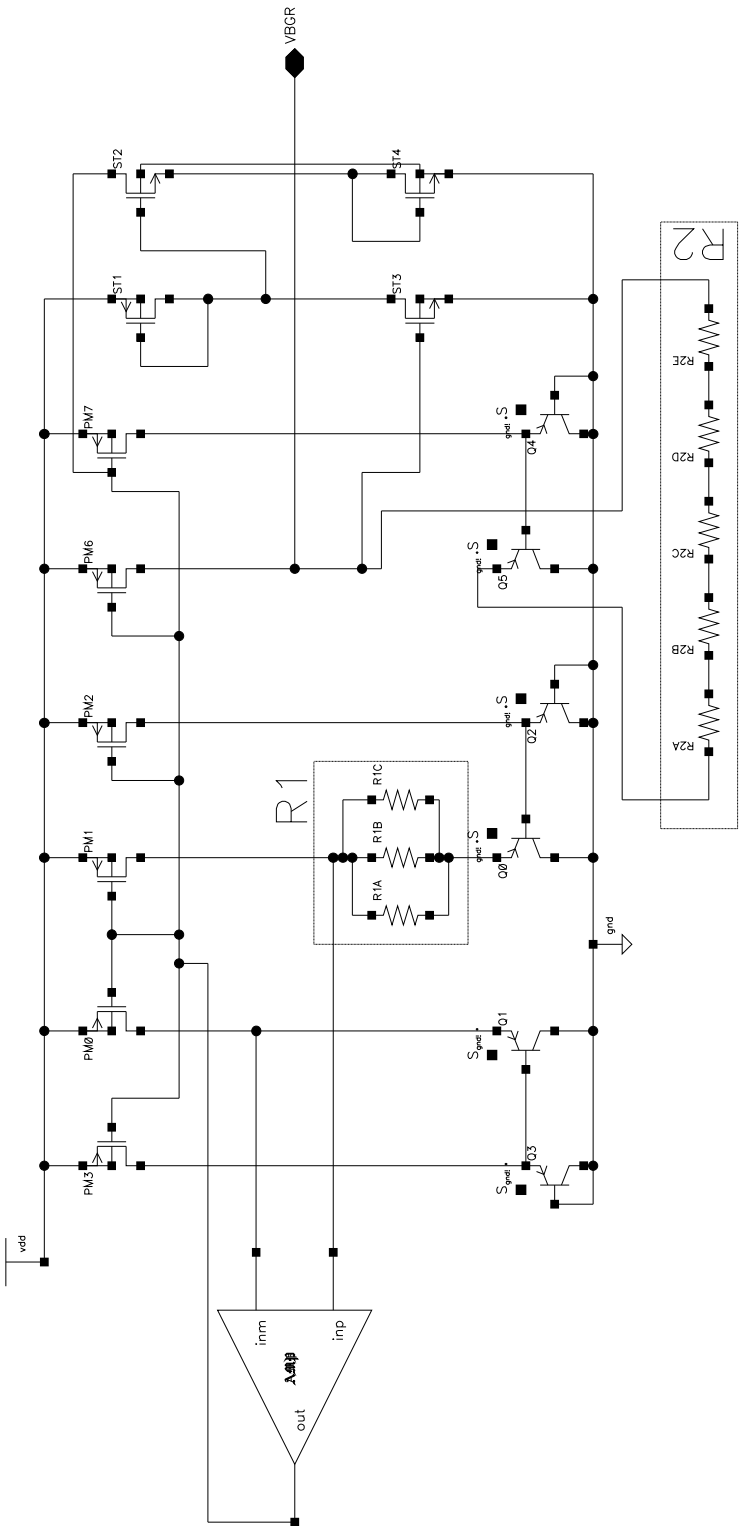



Figure 6.11: The complete BGR schematic

off with $V_{G6}$=VDD and $V_{S6}$=0. Thus transistor $ST_3$ is off and $V_{D\_ST3}$ is near VDD. This causes transistor $ST_2$ to start conducting, which pulls down the gate voltages of all the PMOS current mirrors (i.e., discharging). Eventually they will be coming out of the "0" zone. At this point, $ST_3$ starts conducting current and turns $ST_2$ off. When BGR is in normal operating mode, the start-up circuit should not affect the main circuit. Here $ST_1$ was designed to be a weak transistor to minimize the power consumption.

With regard to the second dynamic issue, its' usually solved by adding a high speed buffer amplifier to decouple the BGR block from the rest of the circuit and improve the response time. The tradeoff here is between speed and power consumption.

### 6.3.5 Layout Design

The importance of judicious layout will never be overemphasized in analog/mixed-signal IC design. In the BGR schematic, the critical matching components include: BJT's $Q_0$ through $Q_6$, PMOS transistors $PM_0$ through $PM_6$, and resistors $R_1$ and $R_2$.

The values of passive components such as resistors and capacitors cannot be controlled precisely in integrated circuits. For N-Well resistors, the resistance variation could be up to $\pm 20\%$ or more. So whenever it is possible, a precision value should always be based on the ratio instead of absolute component value. Fortunately in the BGR design, the accurate resistor value is not utter most important.

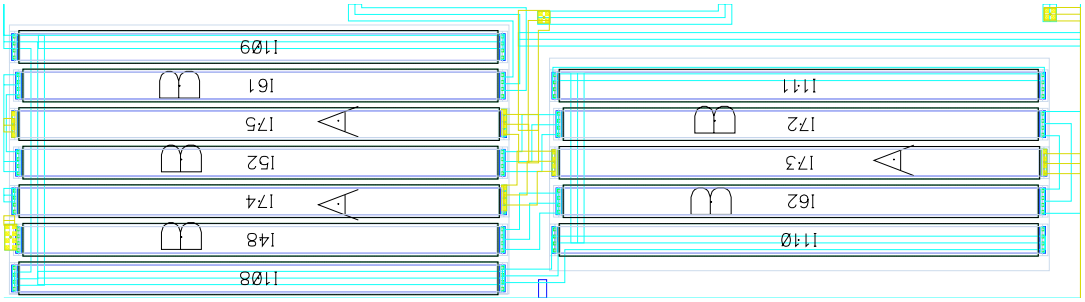



Figure 6.12: Resistor layout arrangement

The point of interest lies mainly in matching the resistor ratios rather than the absolute values. This goal can be achieved through careful layout design. As illustrated in Figure 12, both $R_1$ and $R_2$ were laid out with a $2u$ wide, $36u$ long unit resistor. $R_1$ contains three unit resistors in parallel, while $R_2$ contains five unit resistors in serial. Dummy resistors were placed on the sides to eliminate the uneven etching/doping at the edges. With this arrangement, even if the geometry sizes may deviate from the layout, the resistors ratio will remain the same. Also, using eight resistors instead of two, we have the flexibility to arrange them in a symmetric and common-centroid structure. This helps to average out the process gradient along either direction on the wafer.

Based on the same idea, BJT $Q_1$ and $Q_3$ were used as the unit transistors. $Q_1$, $Q_0$ and $Q_5$ were grouped together, with $Q_1$ placed in the center and surrounded by $Q_0$ and $Q_5$. Similarly $Q_3$, $Q_2$ and $Q_4$ were laid out in another group with the same structure. This arrangement can significantly improve the matching between these BJT's.

The final BGR layout is shown in figure 14. To improve the matching between the current mirrors, the two PMOS transistors $PM_6$ and $PM_7$ were placed in the

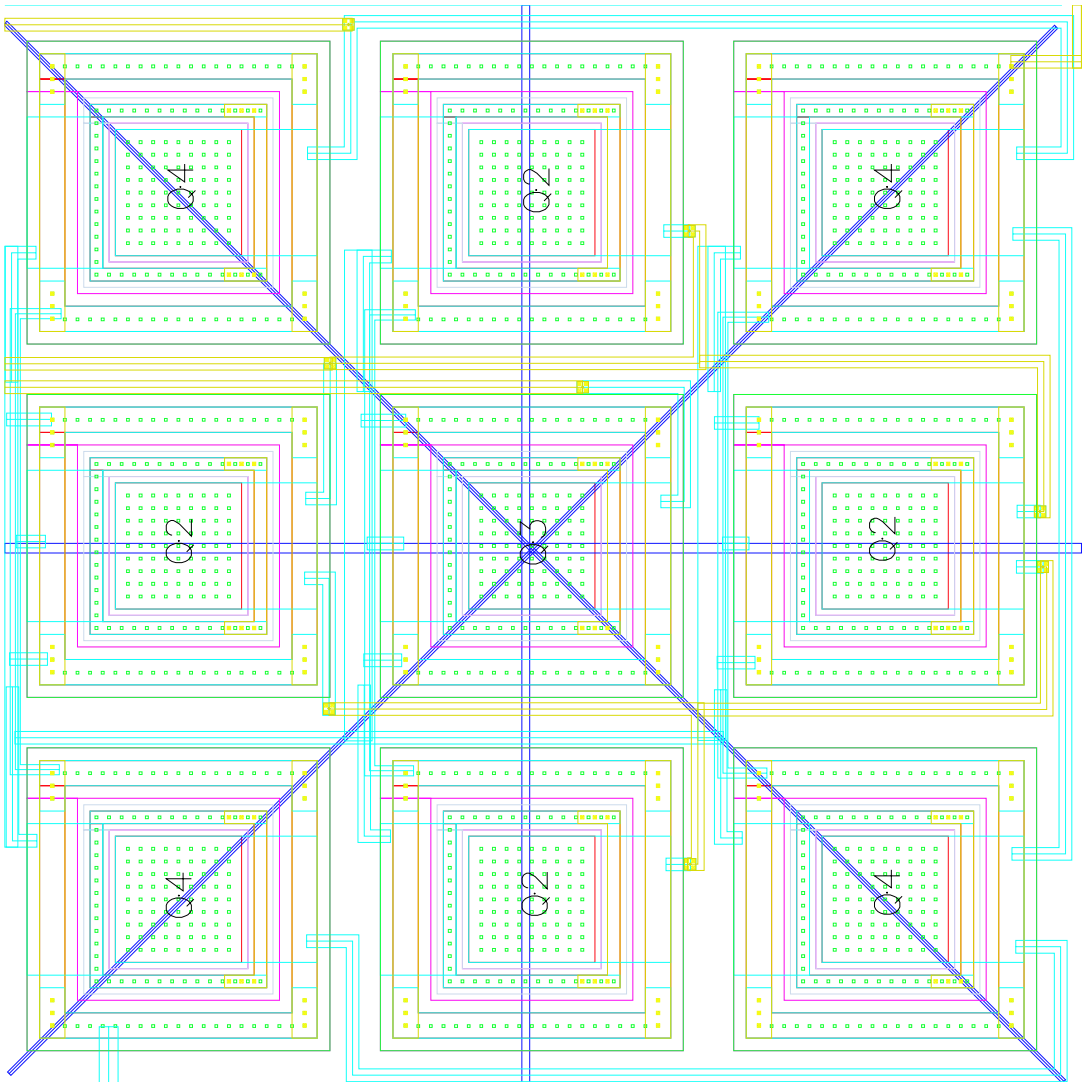



Figure 6.13: BJT layout arrangement

center. Multi-finger structure was used and all the PMOS transistors were splitted into smaller units and placed on the two sides.

### 6.3.6 Results and Discussions

Temperature stability is the primary specification for voltage references. This BGR provides a reference voltage of 2.4927 $V$ at 25°$C$. From 0°$C$ to 85°$C$, the voltage variation is within $\pm 0.587mV$ (figure 15). The maximum TC [4] is 16.06ppm/°$C$ at 85°$C$. It consumes about $1.4mW$ at 25°$C$. It should be noted this design is not optimized for low power operation, but the can be readily modified to significantly reduce the power consumption by using less current branches and low biasing current. Another important specification of BGR is its insensitivity to power supply variation, both at DC and at higher frequency, i.e., AC. For those small especially battery powered devices, the power supply variation may be up to ±10%. Figure 16 shows the BGR output voltage as a function of power supply voltage. The circuit can operate properly at $3V$ with TC of 69.02ppm/°$C$. It's more robust for higher than standard supply voltage. At 3.6 $V$, the maximum TC is 21.7ppm°$C$. If line regulation or cascoded current mirrors are used, the BGR output voltage will be more insensitive to the power supply variation. Figure 17 is the power supply rejection ratio (PSRR) of the BGR. High PSRR can effectively reject the coupling noise from the power supply line. For the noisy environment, BGR wit a higher PSRR can be achieved by using cascoded current mirrors based on the similar concept as discussed in the amplifier chapter. Off-chip decoupling capacitors can also be used

[4] The temperature coefficient here is defined as: $TC = \frac{1}{V_{ref}} \frac{\partial V_{ref}}{\partial T}$.

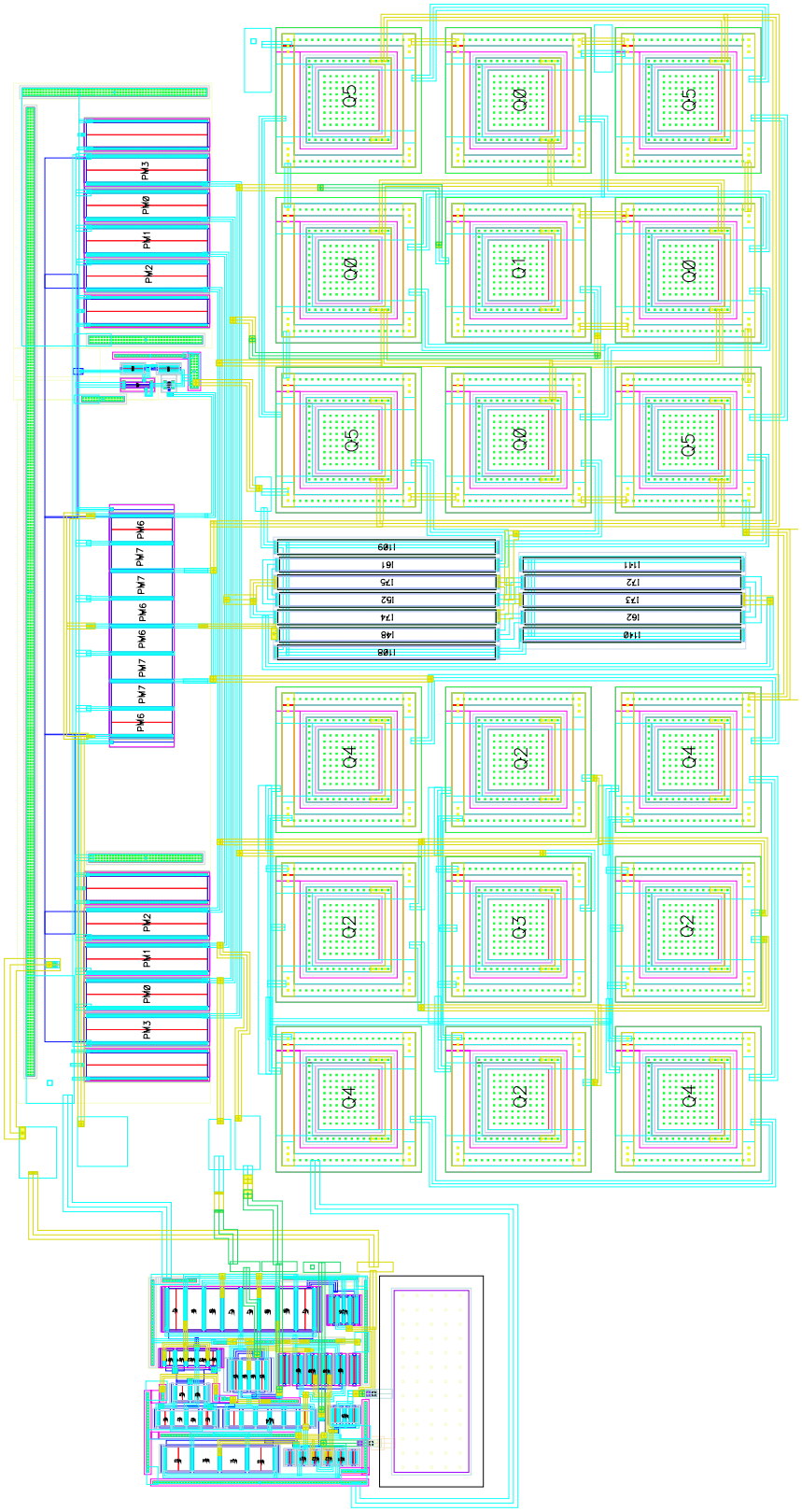



Figure 6.14: Overall BGR Layout

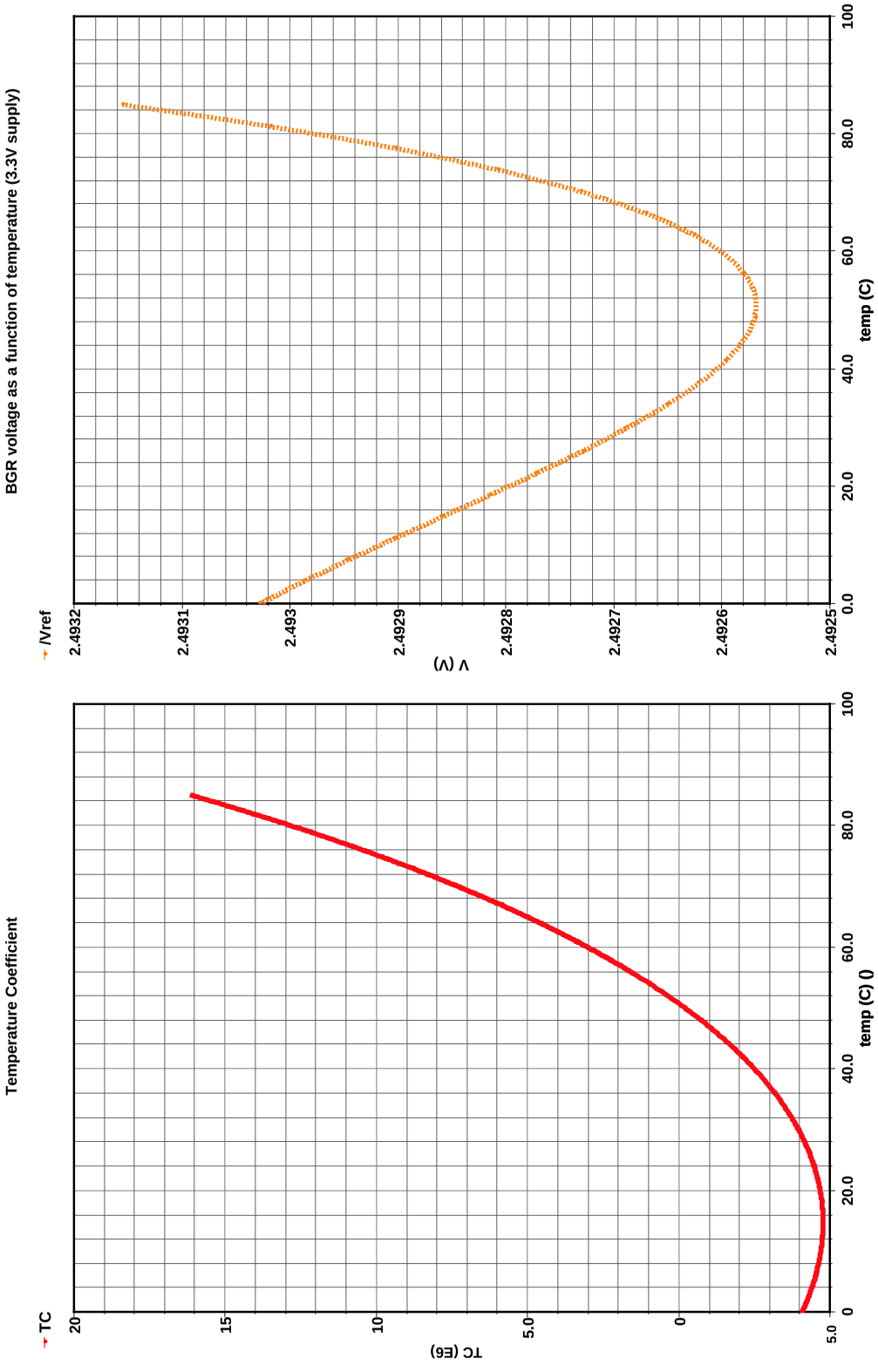



Figure 6.15: BGR Temperature Sweep

to further cancel out the supply noise. The noise performance is a critical specification for low voltage, high precision systems. This BGR circuit was not specifically designed for low noise application. The output noise at 1 KHz is about 10.2 $uV$. The regulation op amp generates most of the noise. By inspecting the amplifier topology again, we can identity that the main noise contributor is from the input differential pair. Because of their smaller sizes, $\frac{1}{f}$ noise is the dominant factor. This has been verified by the simulation. At the cost of more silicon area, the size of two transistors can be increased. Figure 18 is the improved noise performance of this BGR. It shows an output noise of 1.91 $uV$ at 1 KHz with four times the size of original differential pair.

## 6.4 Laser Makelink Trimming for Precision

Many of today's electronics systems are migrating to small footprint and low voltage operation. The reduced supply voltage leaves very small room for errors and increases the accuracy requirements of the reference block. As discussed in the previous sections, the main error contributors in the BGR are the op amp and the resistors. Considering the op amp offset and its finite gain, equation (12) can be re-written as:

$$V_{ref} = 2V_{BE} + \frac{R_2}{R1}(2V_T \ln m + V_{ERR}) \tag{6.14}$$

Due to op amp's high gain, the offset voltage is the main component of $V_{ERR}$. It can be significantly reduced by the laser Makelink trimming method as described in Chapter 5.

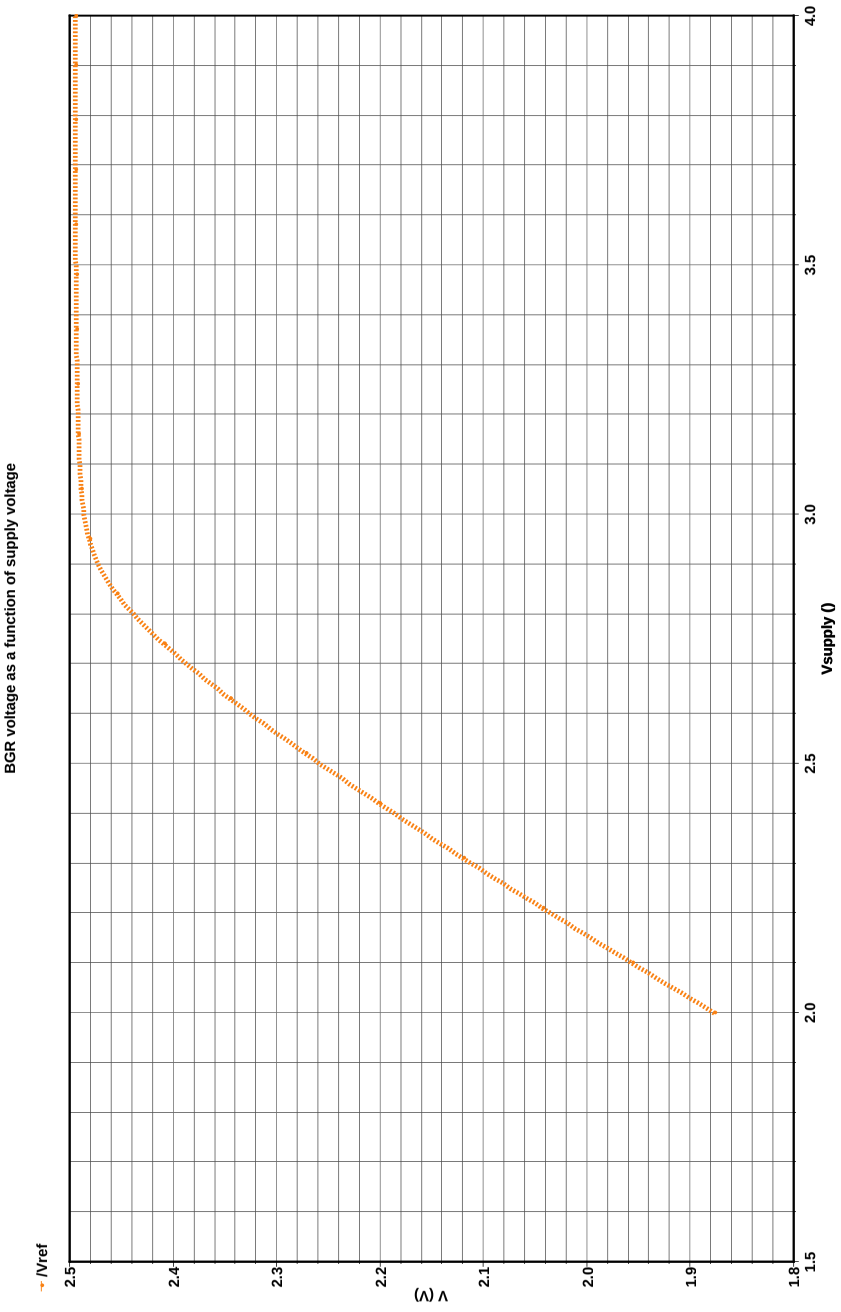



Figure 6.16: BGR voltage as a function of supply voltage

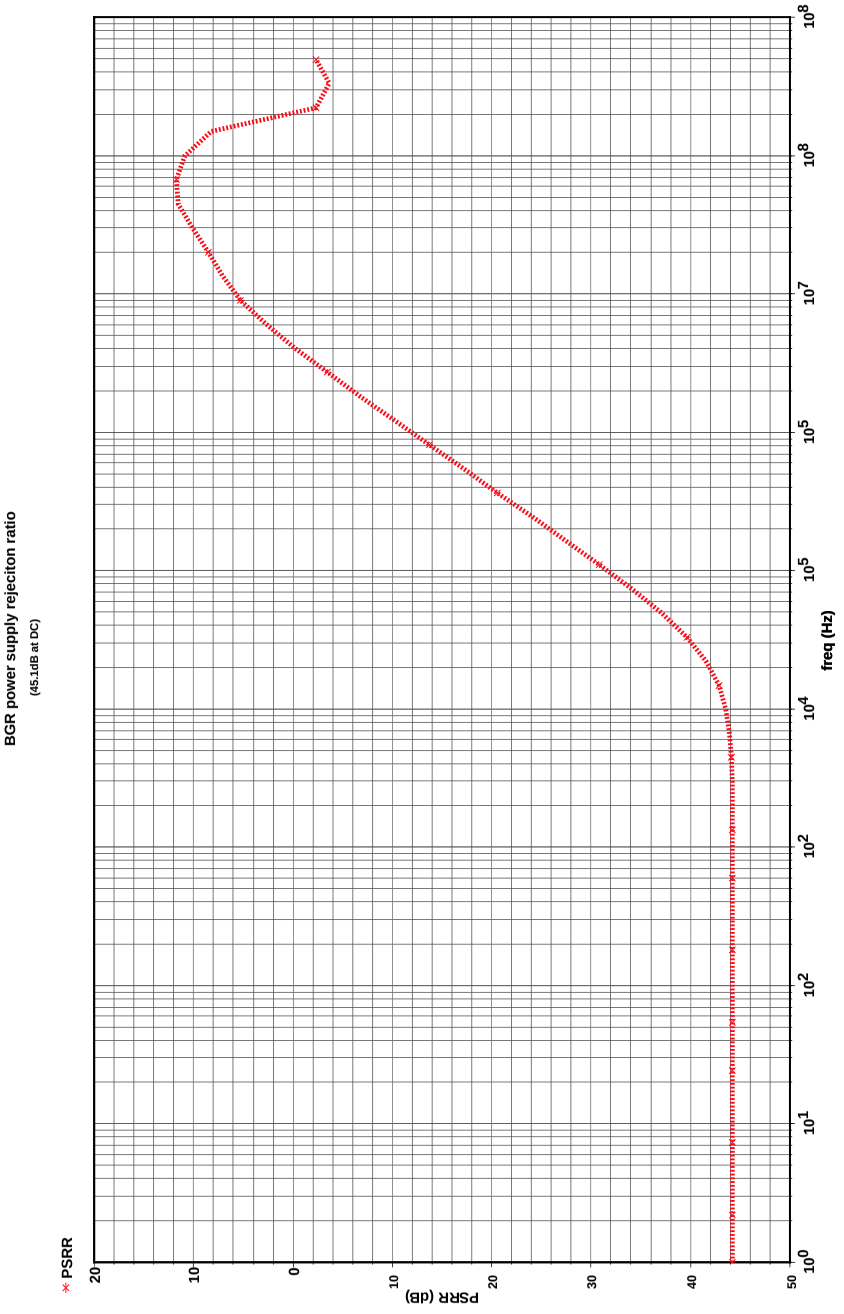



Figure 6.17: BGR power supply rejection ratio

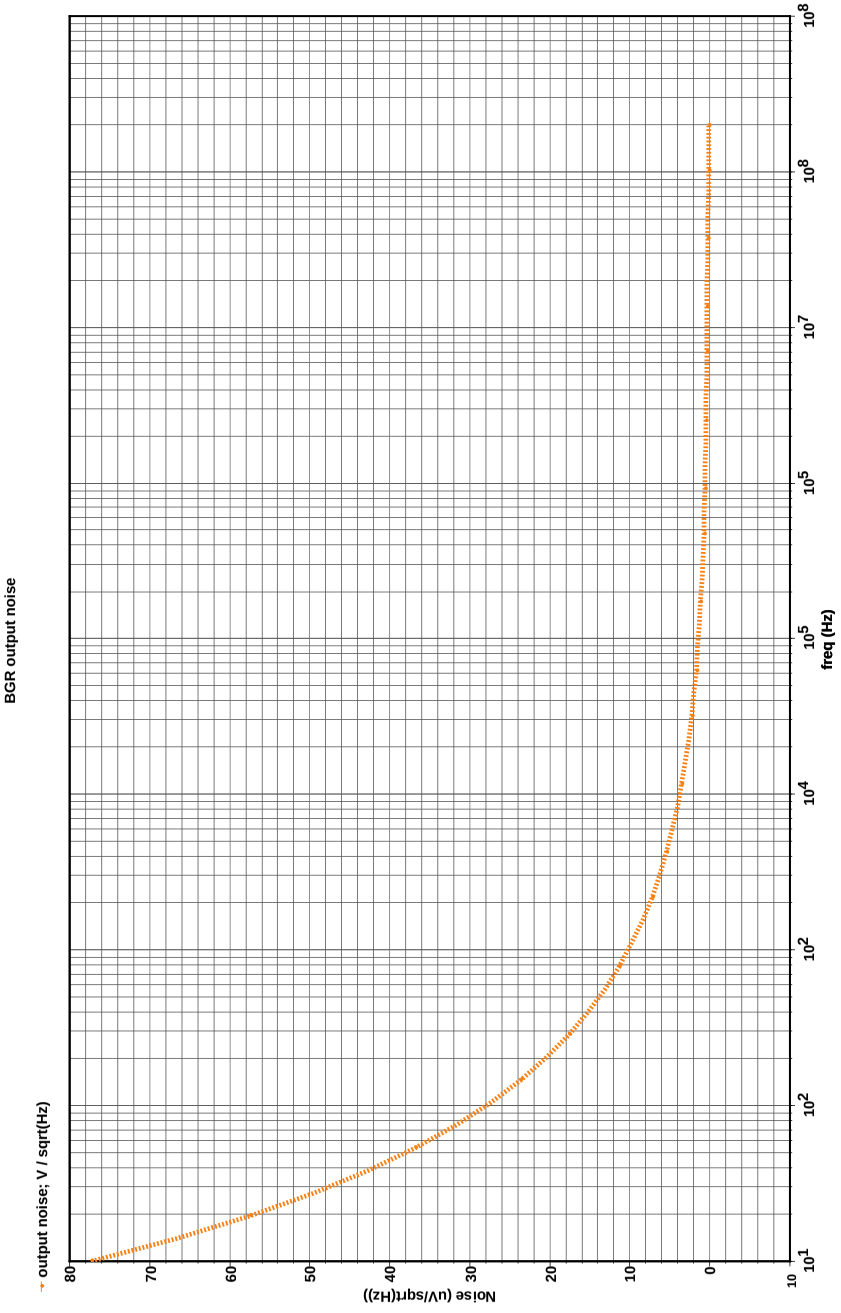



Figure 6.18: BGR output noise

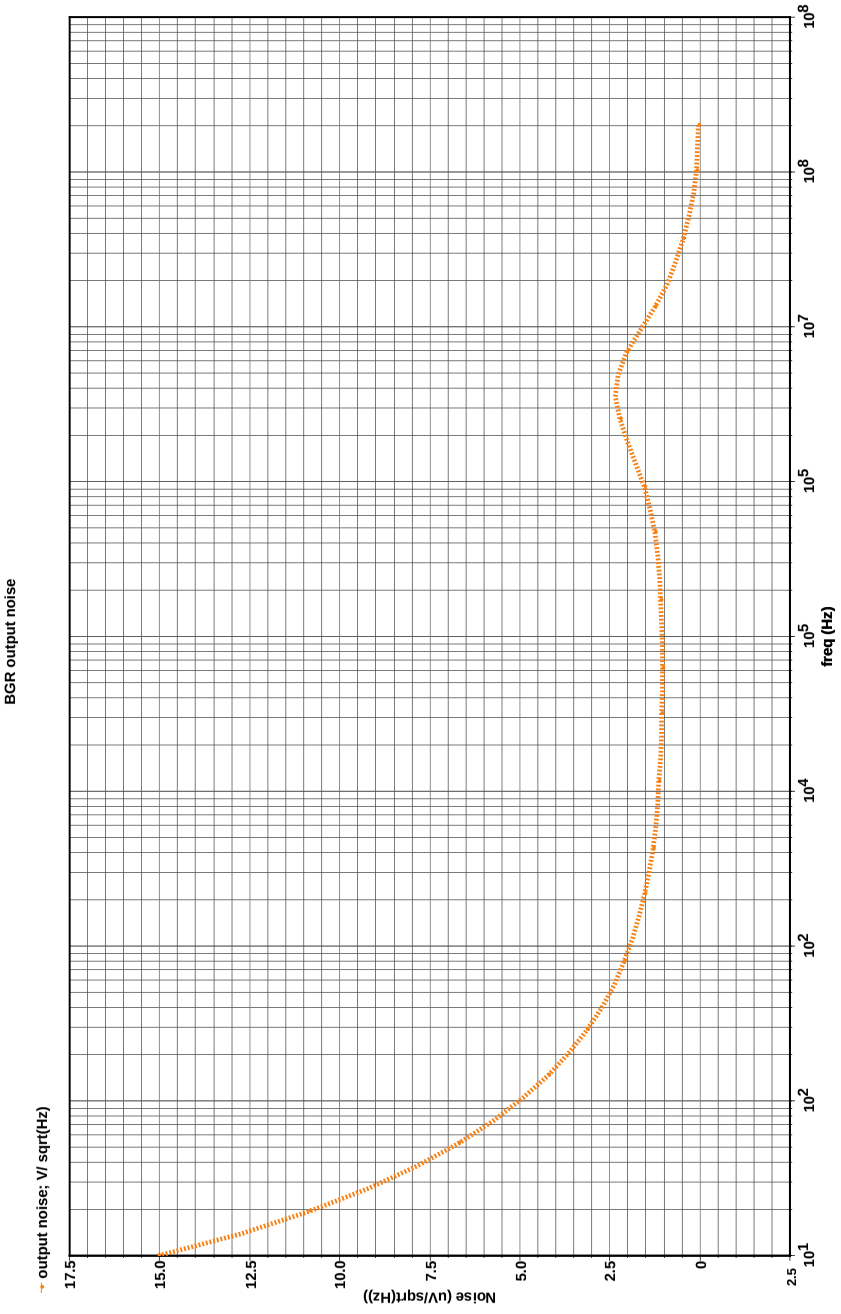



Figure 6.19: BGR output noise with improvement

Another error comes from the resistors. Sometimes even with careful layout design, the process/lithograph induced mismatches is still not tolerable. Laser trimmed thin film resistors are often used for high precision BGR's by many major chip providers such as Texas Instruments, Analog Devices and National Semiconductors. Thin film resistors are very temperature stable and can add to the thermal stability and accuracy of a device, even without trimming. For better accuracy, laser beam are used to "cut" the thin film and very precise resistor values can be obtained. However, the fabrication of this kind of resistors is not compatible with standard CMOS processes. They require the integration of thin film deposition and patterning, which increases the fabrication cost.

Comparing to the traditional laser trimmed thin film resistors, laser Makelink is an excellent alternative. It's fully compatible with all the CMOS processes. It actually forms link instead of "cutting". For this BGR, resistor $R_2$ and $R_1$ can be arranged as follows: In figure 20, $R_N$ is the nominal resistor value. Resistors $R_{T1}$ through $R_{T8}$ are digitally grouped together with minimum value determined by Monte Carlo simulation and process statistics. Using this arrangement, 1-15 times the minimum trimming resistor values can be obtained.

## 6.5 A Low Voltage, Curvature Compensated Bandgap Reference

The design discussed so far is a first order bandgap reference, which should be sufficient for a low to medium resolution system. However, some high precision systems especially those operated at low power supply ($\leq 1.8V$) put a more stringent

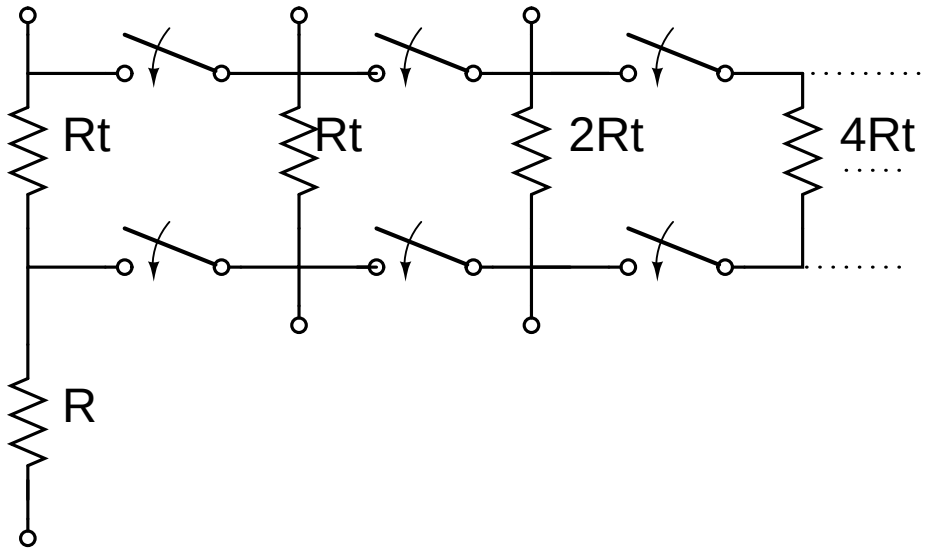



Figure 6.20: BGR programmable resistor for laser Makelink trimming

requirement on the reference accuracy. This mandates the higher order, curvature compensated BGR's.

In the previous BGR design, the temperature dependence of $V_{BE}$ is assumed to be linear. This is only a first order approximation. A more accurate representation of $V_{BE}$ is [91]:

$$V_{BE}(T) = V_{G0} - \frac{T}{T_r}[V_G(T_r) - V_{BE}(T_r)] - (\eta - \delta)V_T \ln \frac{T}{T_r} \tag{6.15}$$

where $V_{G0}$ is the extrapolated bandgap voltage of silicon at 0 K, $T_r$ is the reference temperature, $\eta$ is a process dependent parameter which is usually less than four, and $\delta$ is the order of temperature dependence of the BJT collector current $I_C$. If $I_C$ is PTAT, then $\delta$ is 1. The non-linear temperature dependence of $V_{BE}$ comes from the third term in the equation. It can be further expanded using Taylor series.

Based on equation (15), many creative topologies have been developed to approximately cancel the nonlinear component of $V_{BE}$. A classical method proposed

by Song & Gray is to add a squared PTAT term into the output of the first order bandgap [92]. The basic idea is to cancel out the negative temperature dependence of the logarithmic term in equation (15) with a positive parabolic term. The drawback of this method is a complex circuit is needed to generate the squared PTAT voltage. It occupies more silicon area and consumes more power. Another technique developed by Lee [93] *et. al* is by exploring the temperature dependence of BJT's current gain and exponentially cancel the non-linear component of $V_{BE}$. This is a simple yet very effective technique. However, it is not adequate for low voltage operation because at least a bandgap voltage plus an overdrive voltage are needed.

A CMOS curvature compensated BGR presented in this section was designed based on the topology proposed by Malcovati [94] with some modifications. It can be operated at 1.8 $V$ or even lower power supply. To demonstrate the effectiveness of this method, an BGR without curvature compensation was also designed first. Its schematic is shown in figure 21 (start-up circuit is not included). The same amplifier (figure 8) was used with slight modifications. Comparing to the previous design, this BGR has smaller number of current legs, thus it consumes less power. More importantly, it can be operated at lower power supply as long as $VDD \geq V_{BE} + V_{D,sat}$. The output voltage is defined as:

$$V_{BGR} = \frac{R_3}{R_1}(V_T \frac{R_1}{R_0} \ln N + V_{BE}) \tag{6.16}$$

where $N$ was chosen as 16 so that moderate resistor sizes were used in the layout. The temperature dependence of $V_{BE}$ is canceled out by the first term in the parenthesis to the first order. The output reference voltage may be arbitrarily set by the

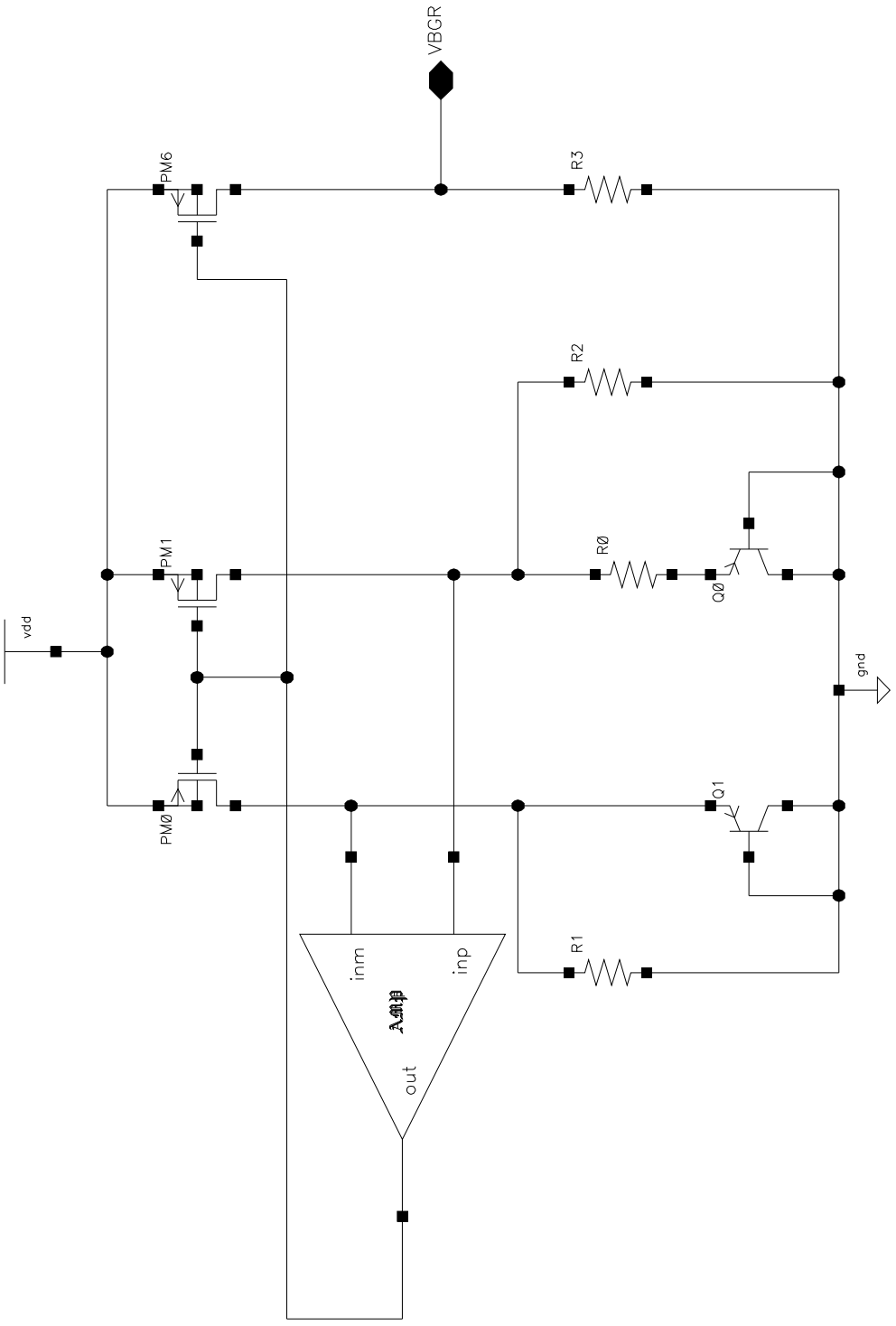



Figure 6.21: A low voltage BGR without curvature compensation

resistor ratio $R_3/R_1$. Thus non-standard value (i.e., $< 1.2\ V$) can be generated. Attention should be paid that it's best to set $V_{BGR}$ to about $0.7\ V \approx V_{BE1} \approx V_{BE0}$ to minimize the current mismatch between the current mirrors.

To further improve the BGR accuracy and reduce its TC, a solution proposed by Gunawan *et. al* [95] was used. The basic idea is to compensate the logarithmic term in equation (15) by a proper combination of an $V_{BE}$ with a temperature-independent current $I_C$ (this implies $\delta \approx 0$) and an $V_{BE}$ with an PTAT current ($\delta \approx 1$). Looking at figure 21, we know the collector currents through BJT $Q_1$ and $Q_0$ are PTAT. Since $V_{BGR}$ is nearly temperature independent, the drain-source current of PMOS $PM_6$ is at first order temperature independent ($\delta \approx 0$). This current can be mirrored and injected into another dioded connected BJT branch. The new curvature compensated BGR is shown in figure 22. Again, using equation (15), the $V_{BE}$ of $Q_0, Q_1$ and $Q_6$ can be expressed as:

$$V_{BE0,1}(T) = V_{G0} - \frac{T}{T_r}[V_G(T_r) - V_{BE0,1}(T_r)] - (\eta - 1)V_T \ln \frac{T}{T_r} \tag{6.17}$$

$$V_{BE6}(T) = V_{G0} - \frac{T}{T_r}[V_G(T_r) - V_{BE6}(T_r)] - \eta V_T \ln \frac{T}{T_r} \tag{6.18}$$

The $V_{BE}$ difference

$$V_{NL} = V_{BE6}(T) - V_{BE0,1}(T) \approx V_T \ln \frac{T}{T_r} \tag{6.19}$$

is a nonlinear term which can be used to cancel the higher order temperature dependence component of $V_{BE}$. The nonlinear current $I_{NL}$ defined by $V_{NL}/R_{4,5}$ is injected into the BGR core. Then the BGR output voltage is:

$$V_{BGR} = \frac{R_3}{R_1}(V_T \frac{R_1}{R_0} \ln N + V_{BE} + \frac{R_1}{R_{4,5}} V_{NL}) \tag{6.20}$$

where $R_4$ and $R_5$ are nominally matched. Because the last term in the parenthesis is used to correct the nonlinear component of $V_{BE}$, it's straight forward to find out that:

$$R_{4,5} = \frac{R_1}{\eta - 1} \tag{6.21}$$

However, the above theoretical analysis cannot be used directly in the actual design because some of inexplicit assumptions were made. First, the $V_{BE}(T_r)$ is not same for BJT $Q_0$ and $Q_6$. Secondly, the resistors have non-zero TC, so $I_{C0}$ is not PTAT and $I_{DS6} \approx I_{C6}$ is not temperature insensitive. Therefore equation (20) and (21) should only be used as a general guidance. The exact resistor values highly depend on the precise process parameter characterization and extensive simulation. For this design, it's found that $\eta - \delta$ is close to 0.5, which is actually much smaller than the expected value of 3. Here, $R_0$ through $R_3$ are the same type P+ Poly resistors. $R_4$ and $R_5$ were intentionally chosen as N + Poly resistors, which have slightly higher negative TC than that of the P+ poly resistors. This makes $R_1/R_{4,5}$ increases as temperature drops. The special choice of resistors proves to compensate the $V_{BE}$ curvature most efficient.

Figure 23 is the comparison between the two BGR's. The curvature compensated BGR clearly shows a significant improvement of accuracy. The maximum TC was reduced from 12.8ppm/°C to 6.21ppm/°C, with maximum reference voltage variation reduced from 297.8 $uV$ to 41.0 $uV$ between 0-85°C. This BGR can work properly with 1.6 $V$ power supply (can work in the sub-1$V$ range with some

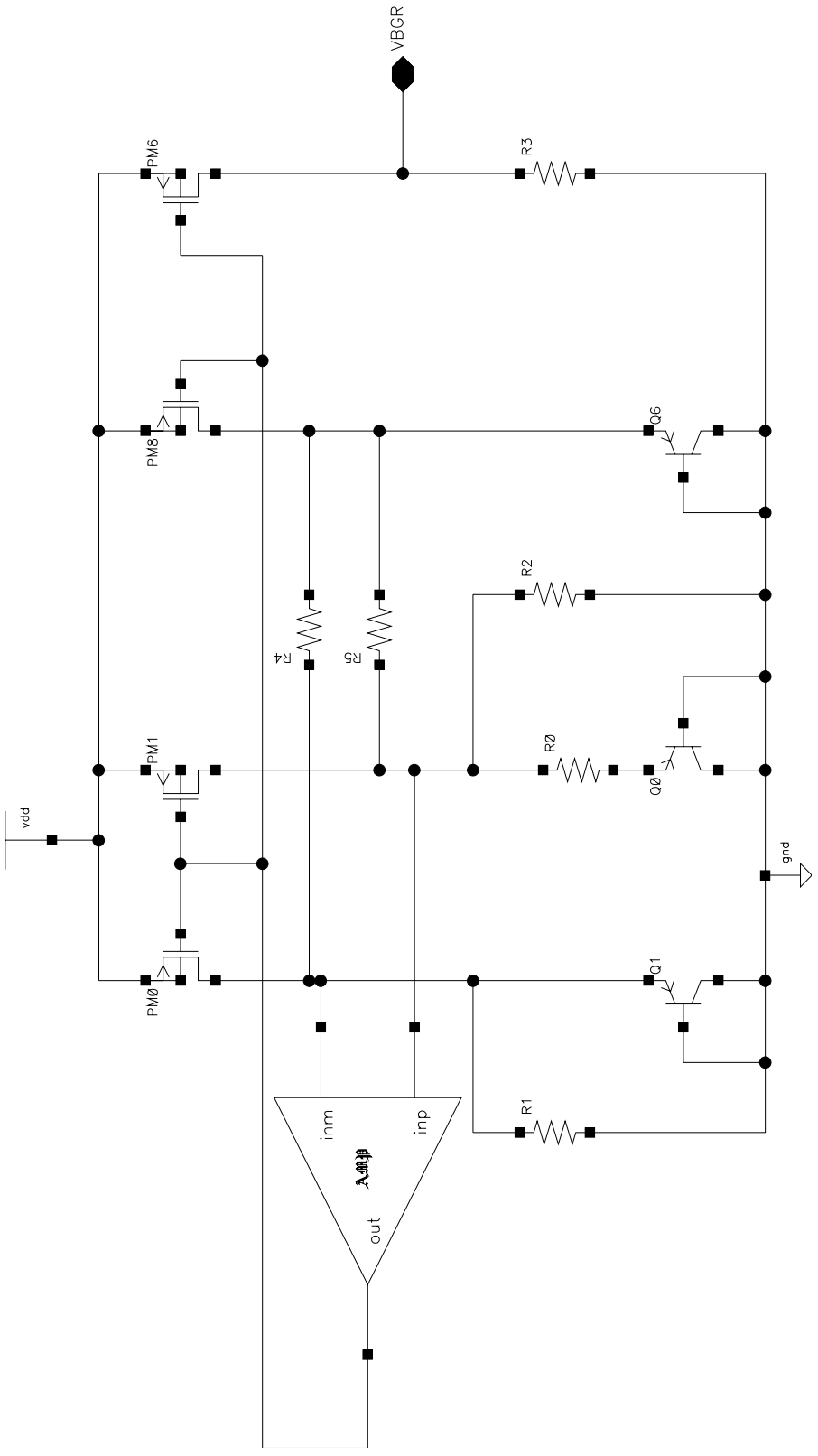



Figure 6.22: A low voltage BGR with curvature compensation

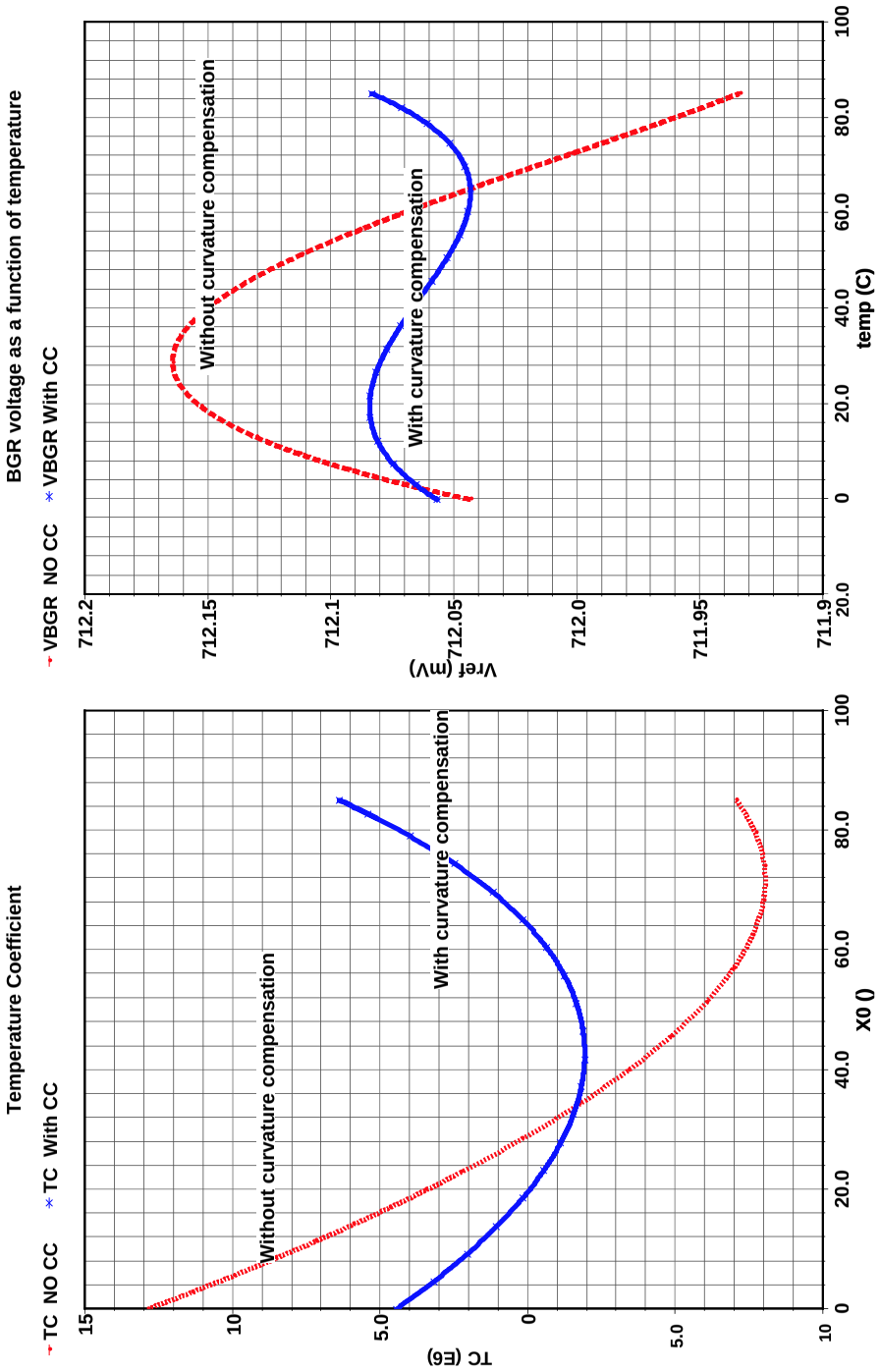



Figure 6.23: Comparison between BGR's with and without curvature compensation

modifications on the op amp). It shows a power supply rejection ratio of 60 $dB$ at DC, and generates a noise voltage of 12.6 $uV/squr(Hz)$ at 1 KHz. It consumes 426 $uW$ power at 27°.

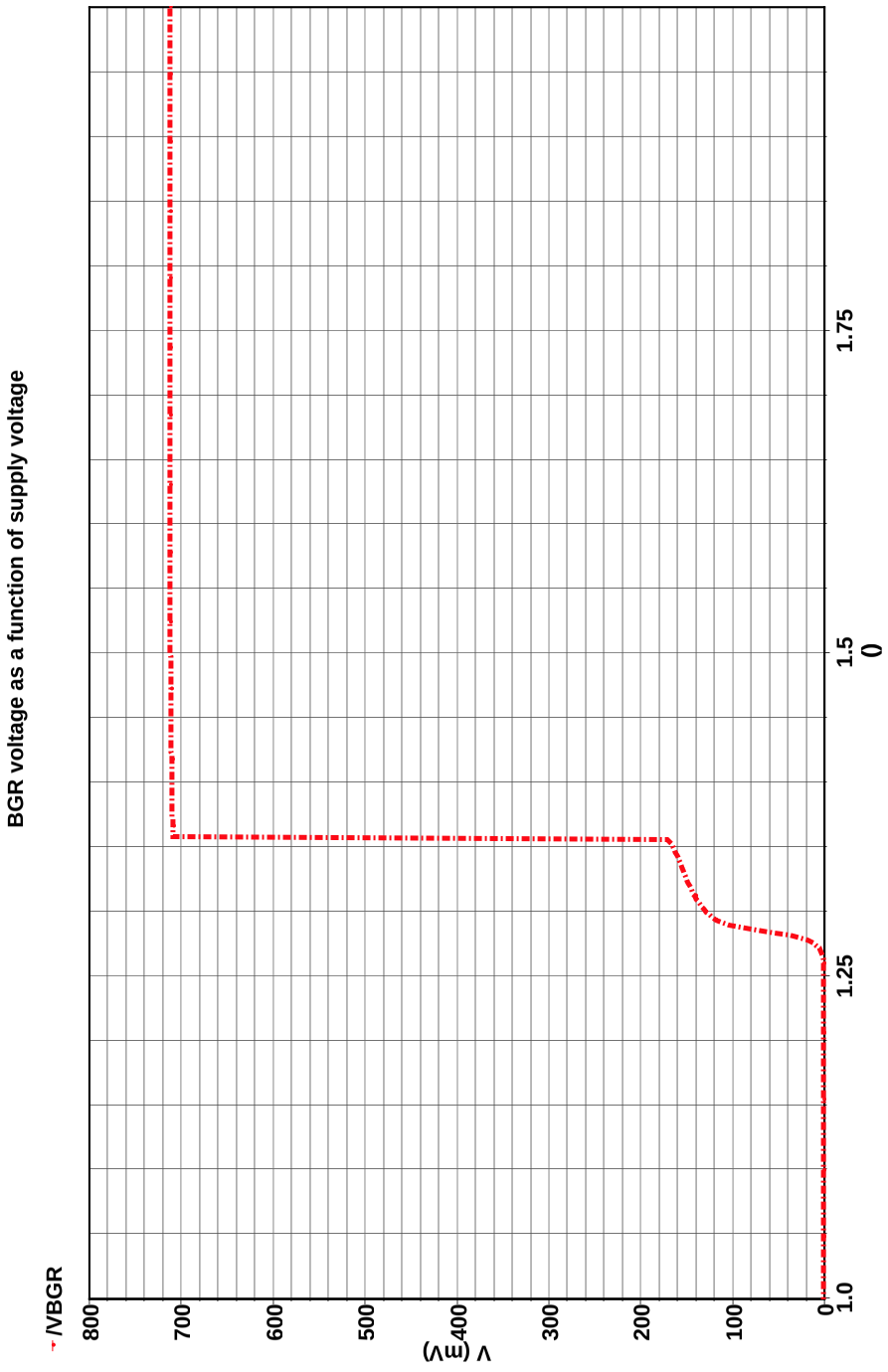



Figure 6.24: BGR voltage as a function of supply voltage variation

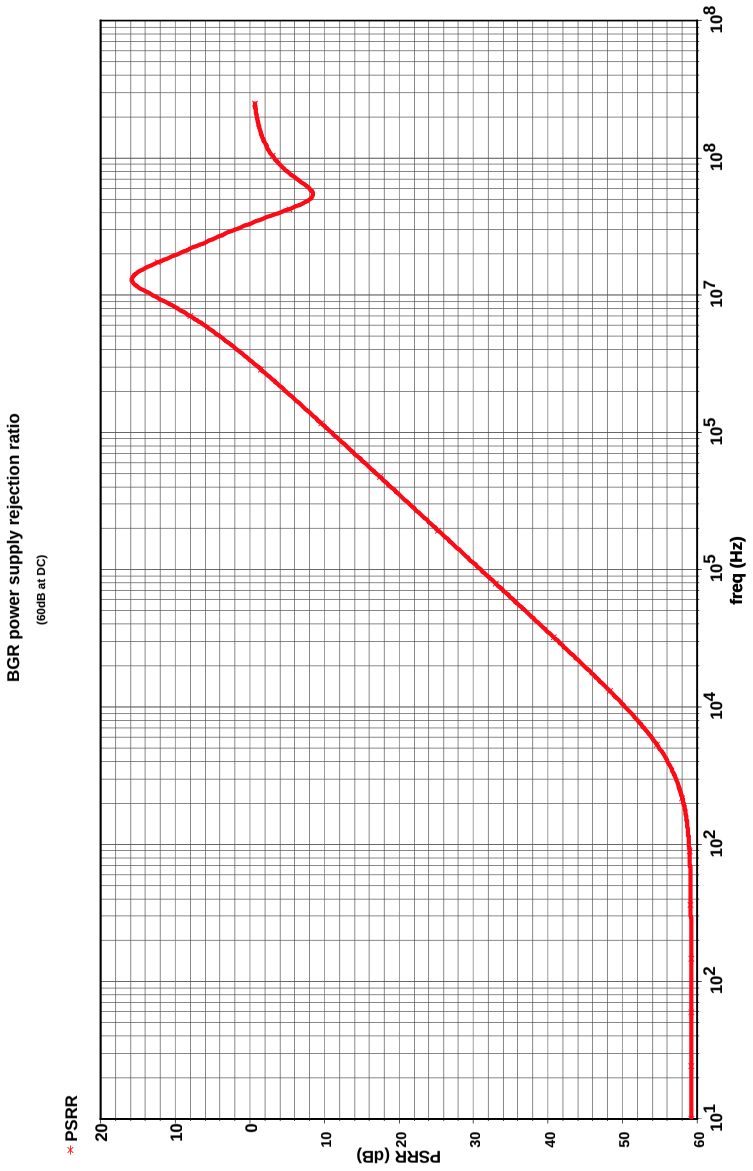



Figure 6.25: BGR power supply rejection ratio

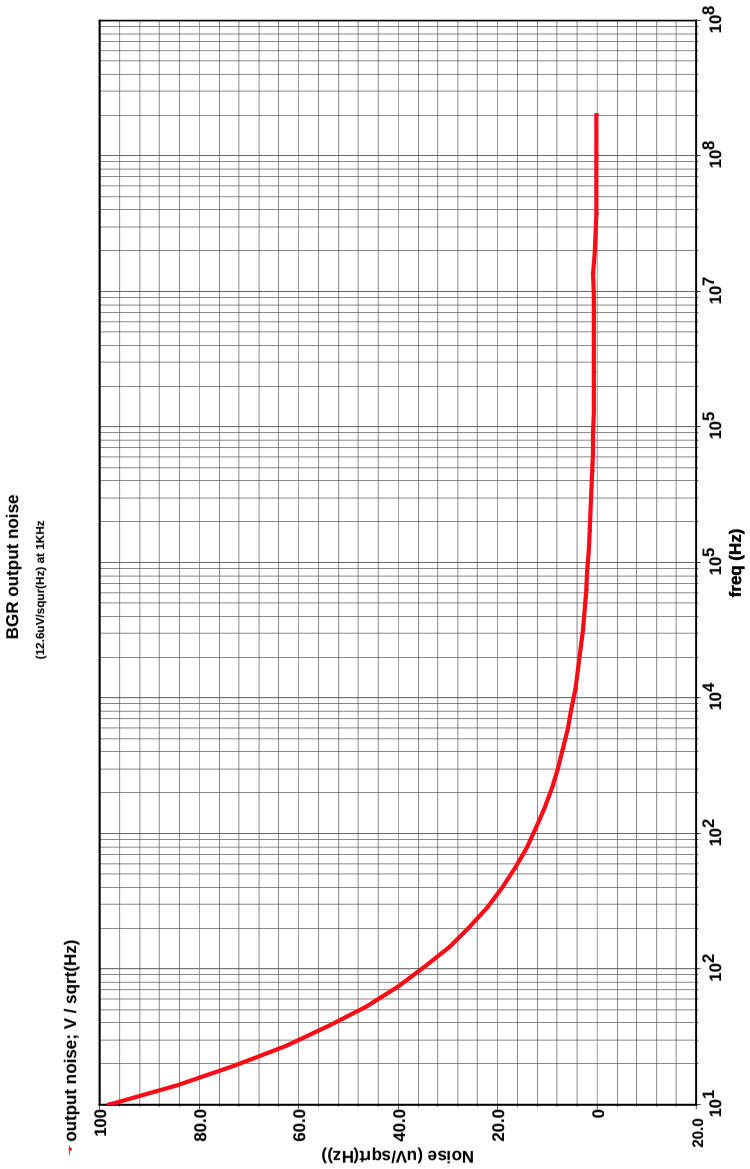



Figure 6.26: BGR noise performance

# Chapter 7

# FPAA Applications

## 7.1 CAB Based Applications

The differential difference op amp (DDA) is a powerful analog circuit building block. The amplifier itself combined with some passive components can implement many analog signal processing functions. This section devotes to some CAB based applications using this amplifier and the CAB architecture developed in the previous chapter. All the circuits employ fully differential topologies.

### 7.1.1 Gain Amplifier

The straight forward applications of this op amp would be various gain amplifiers, either inverting or non-inverting. Probably the easiest and most widely used configuration is the fully differential unity-gain buffer, as shown in figure 7.1 (a). Comparing to the standard implementations which require four matched resistors with one 2-input, 2-output differential op amp, or two matched, single ended op amp, this DDA based implementation is simpler and more accurate. Figure 7.1 (b) and (c) are the inverting and non-inverting gain amplifier configurations. It should be noted though, in contrast to the inverting configuration, the non-inverting gain am-

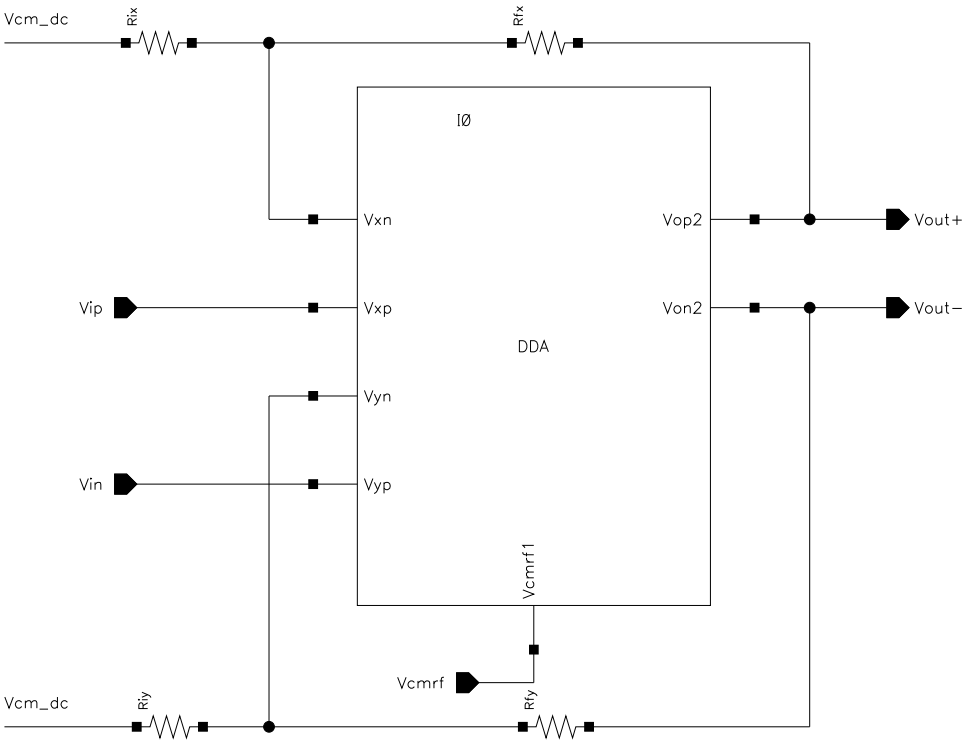



Figure 7.1: Non-inverting gain amplifier configuration

plifier displays a high input impedance that doesn't load previous stage. For voltage mode operation, it is usually desirable to use the non-inverting amplifier whenever it's possible. The typical frequency response is shown in figure 7.2 for different gain values. A differential transresistance amplifier can also be easily implemented with one DDA, as shown in figure 7.1(d).

Figure 7.3 (a) demonstrates another DDA application, voltage-controlled-current-source (VCCS). The output current is determined by $V_{in}/R$. Figure 7.3 (b) shows a sinusoidal voltage controlled current source, $500mV$ over a $10K$ resistor. In fact, the control voltage doesn't have to be an AC signal. A stable voltage source, such as a bandgap can be fed into the op amp as the control voltage (through a simple single to differential ended conversion). If the loading resistors are chosen to be the

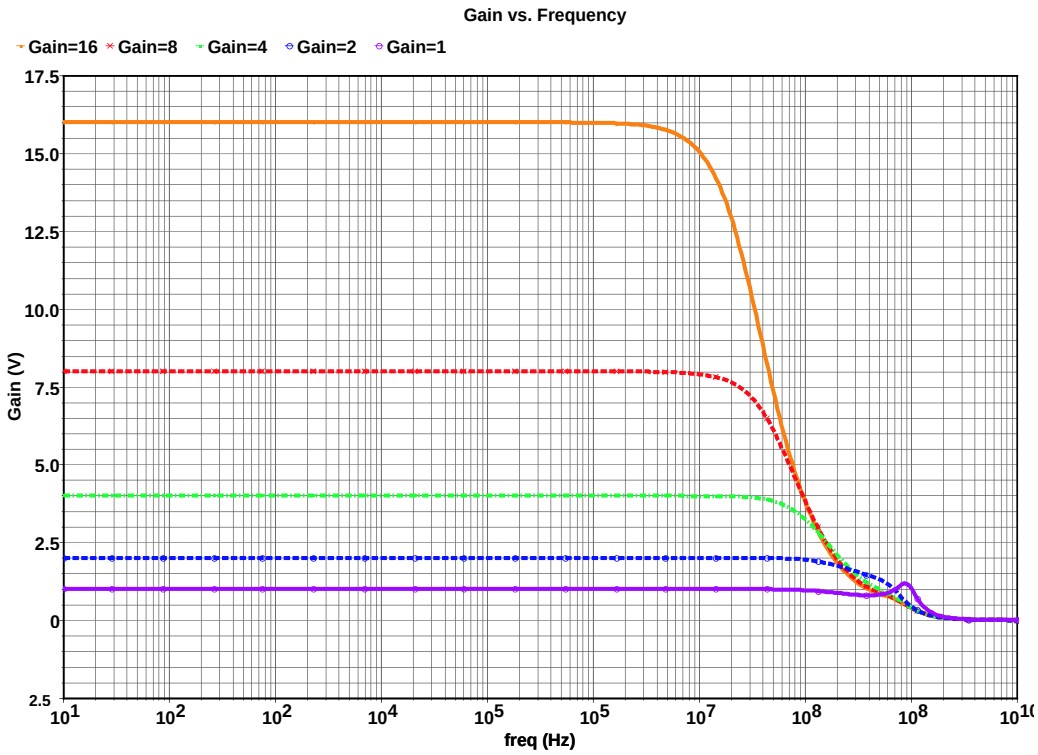


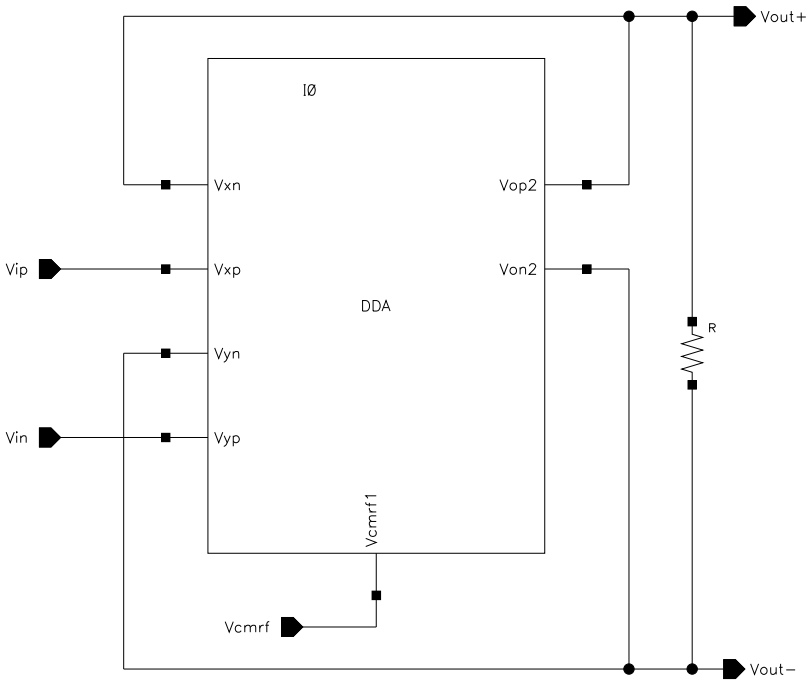

Figure 7.2: Non-inverting gain amplifier frequency response

same type of resistors, then this configuration can be used to generate the reference voltages for ADCs or DACs, as shown in figure 7.4.

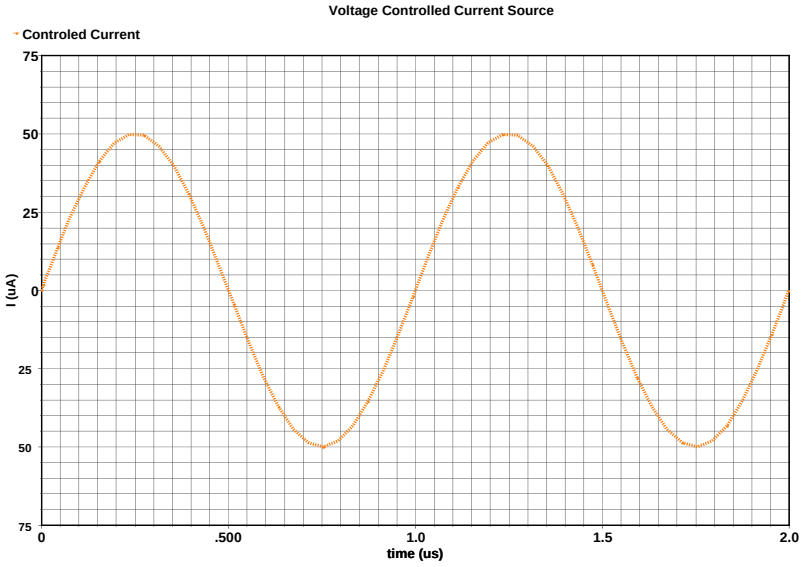
DDA together with two MOS transistors operated in triode region can also be used as a modulation/multiplication cell, as shown in figure 7.5 (a). Two same size PMOS transistors were used because their source and body can be shorted together to reduce the signal dependent nonlinearity. When they are biased in the triode region, the source drain current follows a linear relationship:

$$I_{DS} \approx 2\beta(V_S - V_G - V_{thp}) \tag{7.1}$$

where $\beta$ is the transconductance parameter, $V_G$ and $V_S$ are the source and gate voltage, respectively. In the above schematic, the carrier signal $V_c = v_{cp} - v_{cn}$ have




(a)




(b)

Figure 7.3: Voltage controlled current source (VCCS) (a)schematic; (b)output

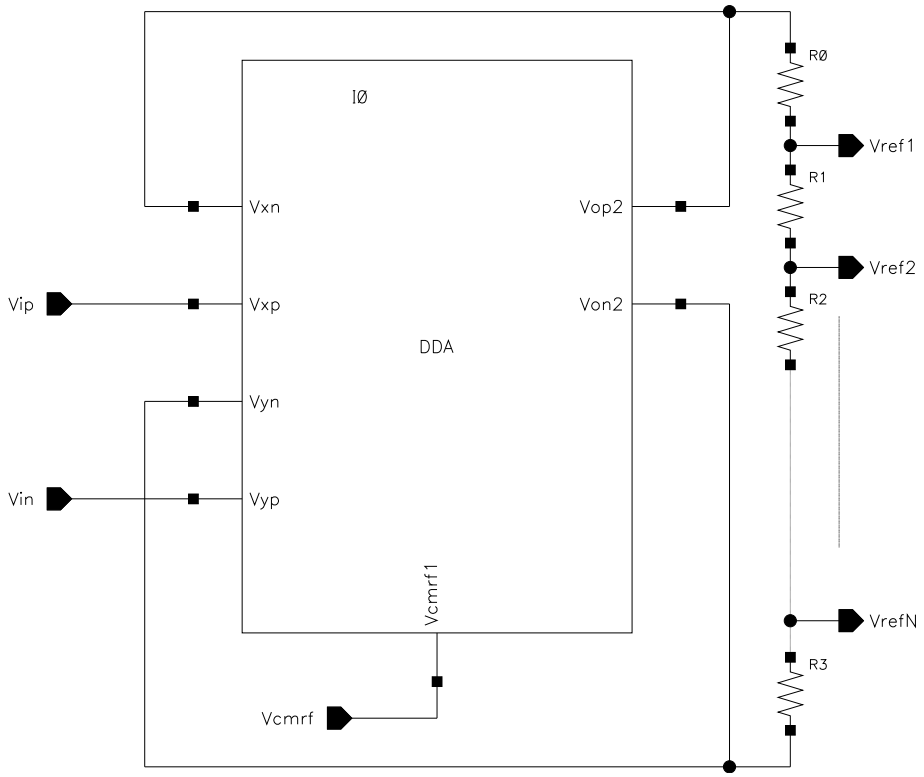



Figure 7.4: A reference voltage generation block for ADC

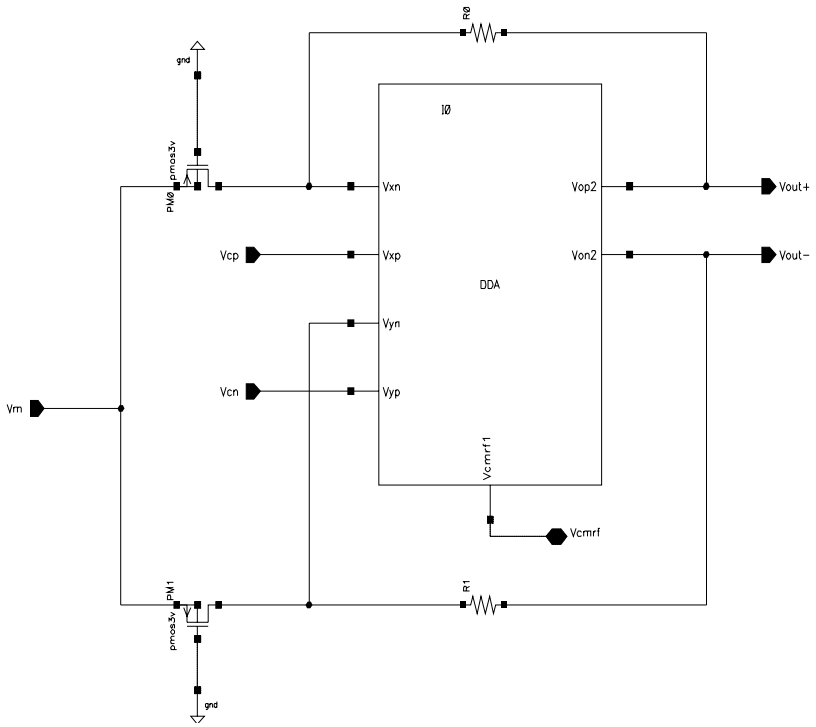
the same common-mode DC level as the modulating signal $V_m$. The gates of the PMOS transistors are biased at 0. The modulated output is given as:

$$V_{out} = 2[1 - 2\beta(V_c - V_{thp})]V_m \tag{7.2}$$

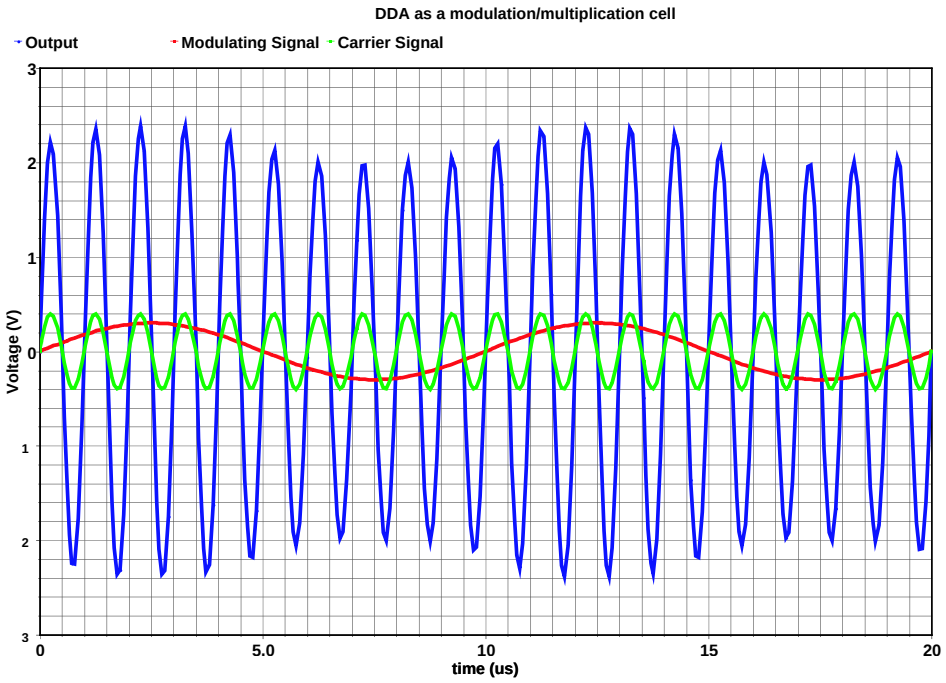
Figure 7.5 (b) shows the result.

### 7.1.2 Active Analog Filter

Filters are the fundamental building blocks in all kinds of analog signal processing systems. They can be categorized into discrete analog passive filters, switched-capacitor filter, active analog filters (including RC active filter and $G_m$-C /MOSFET-C filters), passive LC filters and distributed (waveguide) filters. Figure 7.6 summarizes the choice of filter type based on the desired operating frequency [96]. One




(a)




(b)

Figure 7.5: DDA as a modulation/multiplication cell (a)schematic; (b)output

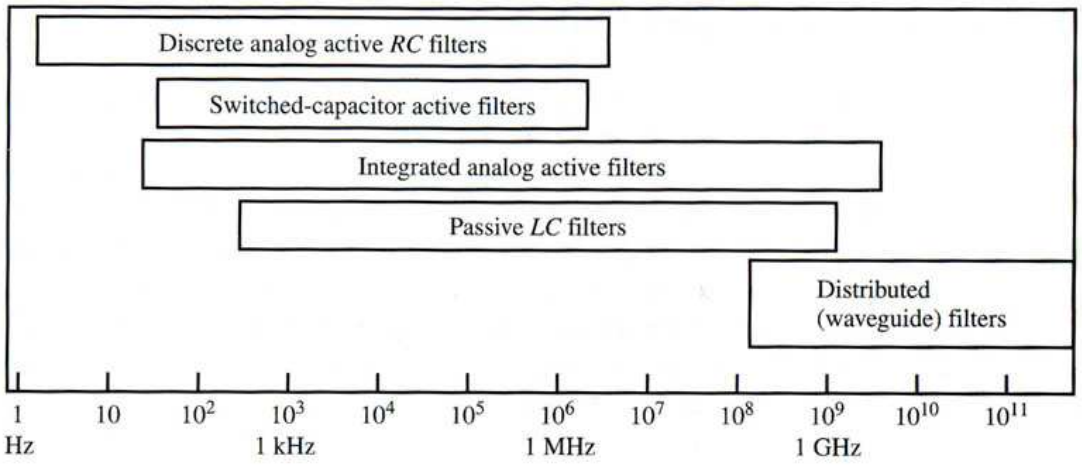



Figure 7.6: Choice of filter as a function of the operating frequency range

of the motivations of this work is to develop a continuous-mode operated FPAA suitable for high frequency operation. Although switched-capacitor filters have the advantage of less sensitive to the component precision, its bandwidth is usually limited to $1MHz$. So only active RC filters are discussed here. It should be noted though the DDA op amp itself has no problem to to used for either type of the filters.

When it is internally compensated with larger than 45° phase margin, the DDA can be modeled as a first order system with transfer function of:

$$H(s) \approx \frac{\omega_u}{s} \tag{7.3}$$

where $\omega_u$ is the unity-gain frequency of the amplifier. As a general rule of thumb, when used in an active filter, the bandwidth of the op amp should be at least 10 times of the filter's cut-off frequency [97], because as frequency goes up, the op amp's dominant pole is coming to play thus there's more "unexpected" roll-off. One

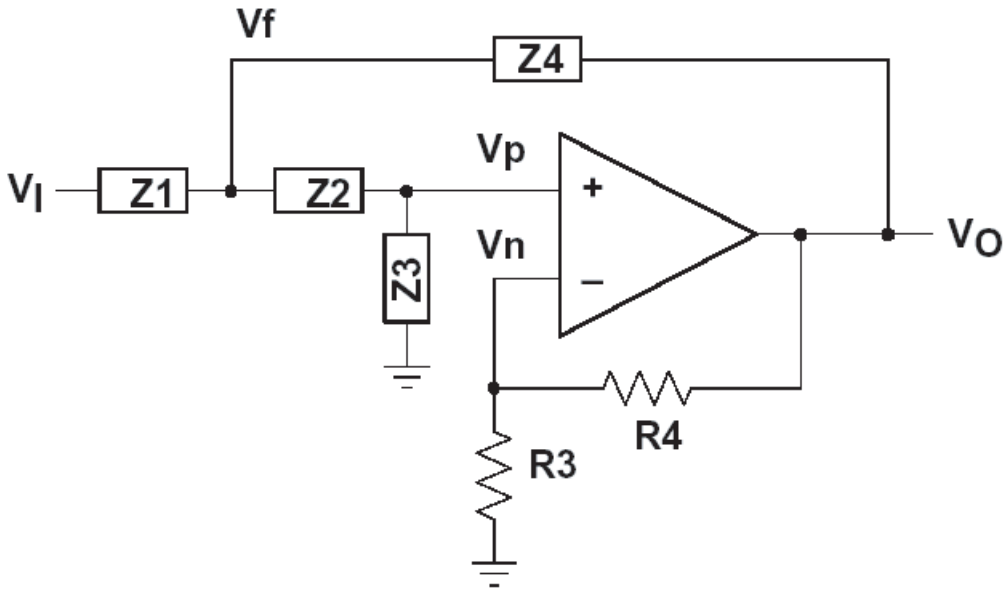



Figure 7.7: Generalized Sallen-Key topology

exception is that for some low pass filters, this "extra" roll-off may be welcome since it provides more attenuation.

The Sallen-Key structure [98] is a popular filter implementation method. It only requires one op amp per bi-quad. Thus it's simple and especially attractive when cost and power consumption are concerns. Figure 7.7 is a general representation of this topology. The voltage transfer function is given as:

$$\frac{V_o}{V_i} = \frac{K}{\frac{Z_1 Z_2}{Z_3 Z_4} + \frac{Z_1}{Z_3} + \frac{Z_2}{Z_3} + \frac{Z_1(1-K)}{Z_4} + 1} \tag{7.4}$$

where $K = 1 + R_4/R_3$ is the filter gain. By properly choosing the component types and values, low pass, high pass or bandpass filter response may be realized. Using the DDA developed in chapter 5, fully differential Sallen-Key filter can be readily implemented.

It's usually difficult to design a fully differential Sallen-Key bandpass filter

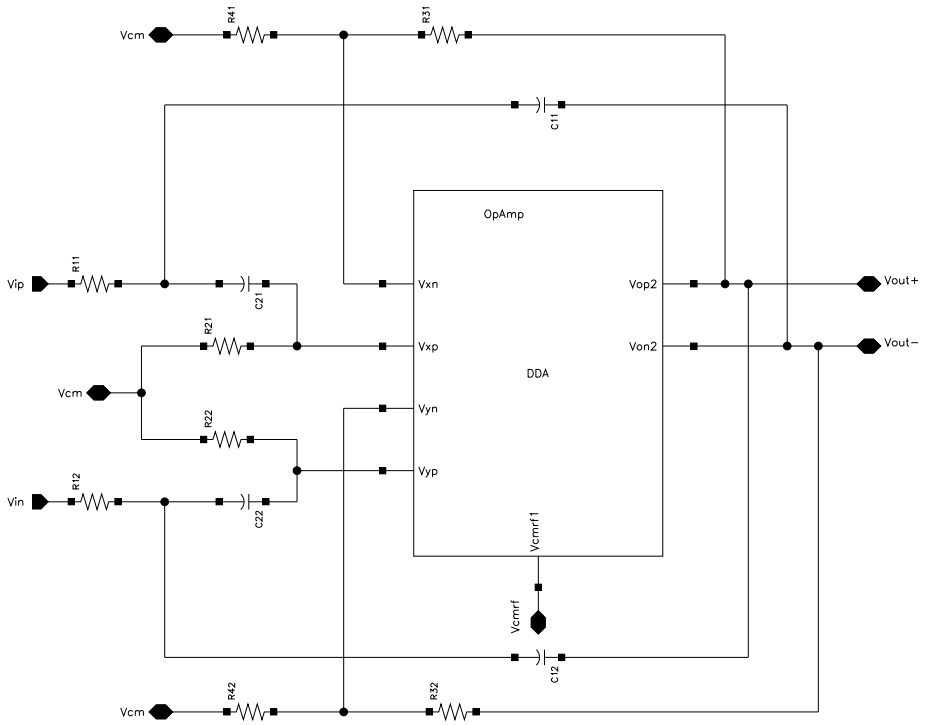
using one standard op amp. The DDA provides an easy solution [99]. Figure 7.8 (a) is the implementation schematic. The center frequency and the quality factor can be expressed as:

$$\omega_0 = \sqrt{\frac{1}{C_1 C_2 R_1 R_2 (1+k)}} \tag{7.5}$$

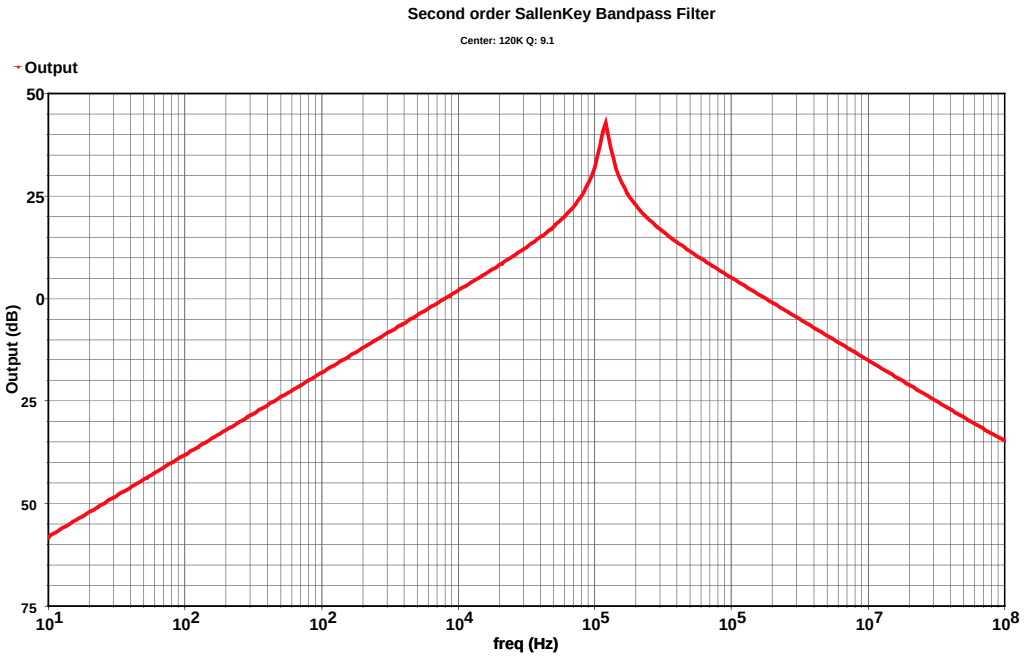
$$Q = \frac{\sqrt{C_1 C_2 R_1 R_2 (1+k)}}{C_1 R_1 + C_2 R_2 + C_2 R_1} \tag{7.6}$$

where $k = (1+\frac{R_3}{R_4})$ is the gain. While this implementation is simple and less sensitive to the component values [99], but since the quality factor is directly related to the gain, so it's difficult to adjust them independently. Moreover, the high $Q$ have to achieved by high gain with wide bandwidth. This may mandate less compensation thus bring the risk of instability. Figure 7.8 (b) shows the simulated result.

It is also very convenient to implement low-pass and high-pass Sallen-Key filters using DDA and some passive components. Filter 7.9 (a) is a third order Butterworth low-pass filter implemented by cascading a first order stage with a second order Sallen-Key. Figure 7.9 (b) is the filter's frequency response. It shows a $-3dB$ cut-off at $10.3MHz$ with $60dB/decade$ roll-off in the transition band. For Butterworth filter, there's no ripple in the passband. Even though the component value may not be precisely controlled, this filter can still be practically used on-chip as an anti-aliasing filter for some high speed, low to medium resolution data converters (for examples, $SNR \approx 60dB$), since the attenuation at the aliasing frequency is already below the noise floor. The transfer function of this filter can be expressed

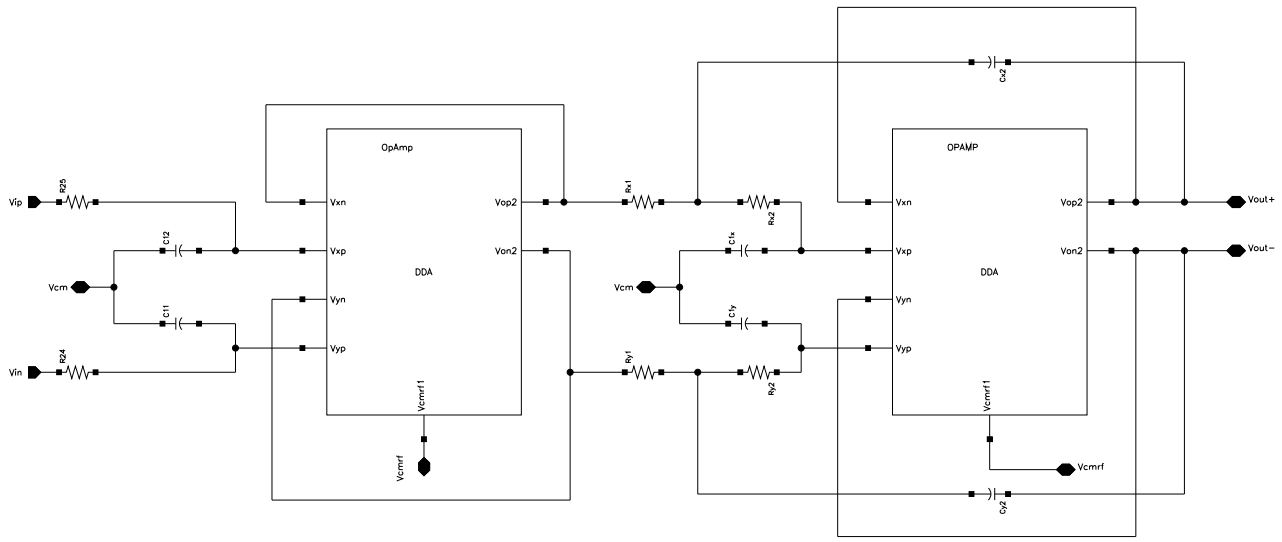



(a)

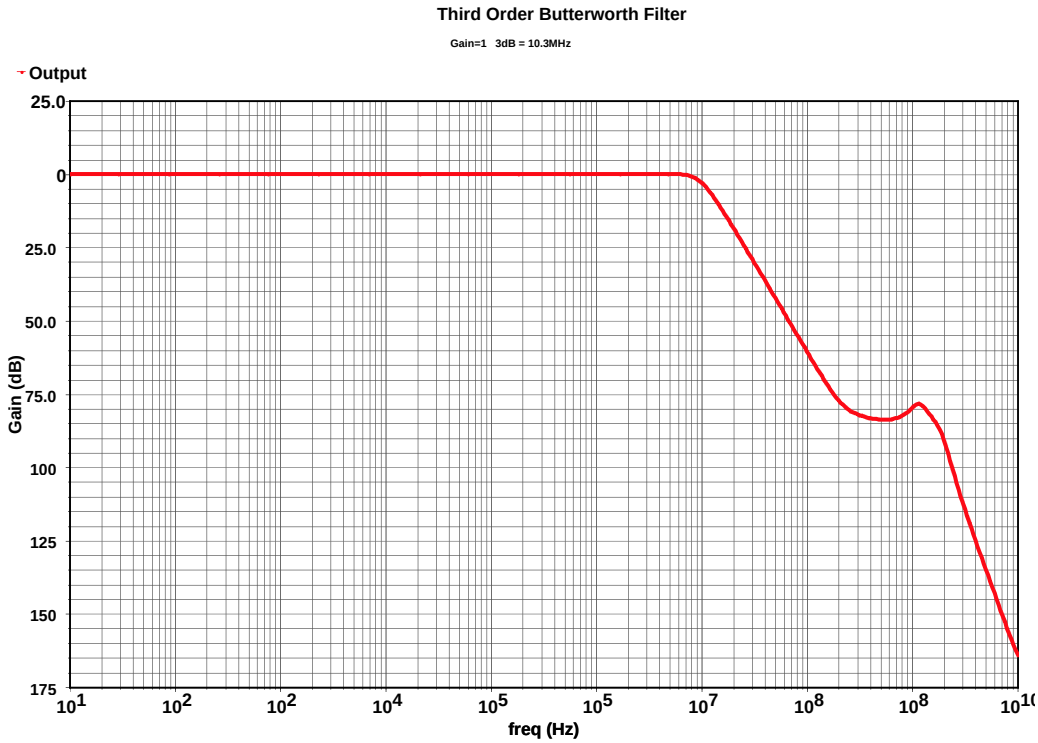



(b)

Figure 7.8: A second order Sallen-Key narrow band-pass filter




(a)




(b)

Figure 7.9: A third order Butterworth low-pass filter based on Sallen-Key topology

as:

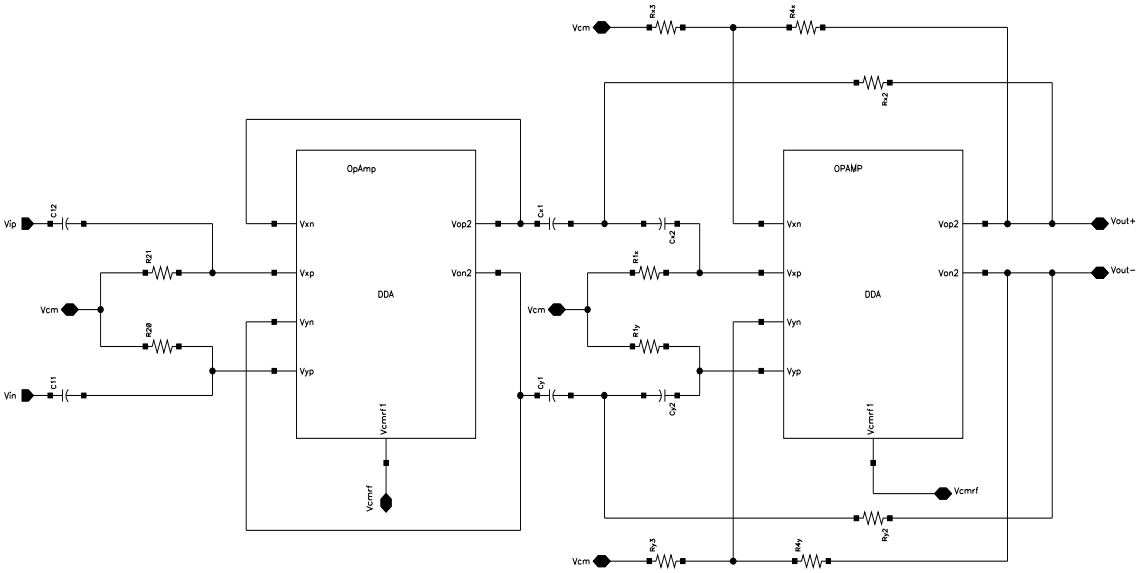
$$\frac{V_o}{V_i} = \frac{K}{\frac{Z_1 Z_2}{Z_3 Z_4} + \frac{Z_1}{Z_3} + \frac{Z_2}{Z_3} + \frac{Z_1(1-K)}{Z_4} + 1} \tag{7.7}$$

Similarly, a third order Butterworth high-pass filter can also be implemented, as shown in figure 10.

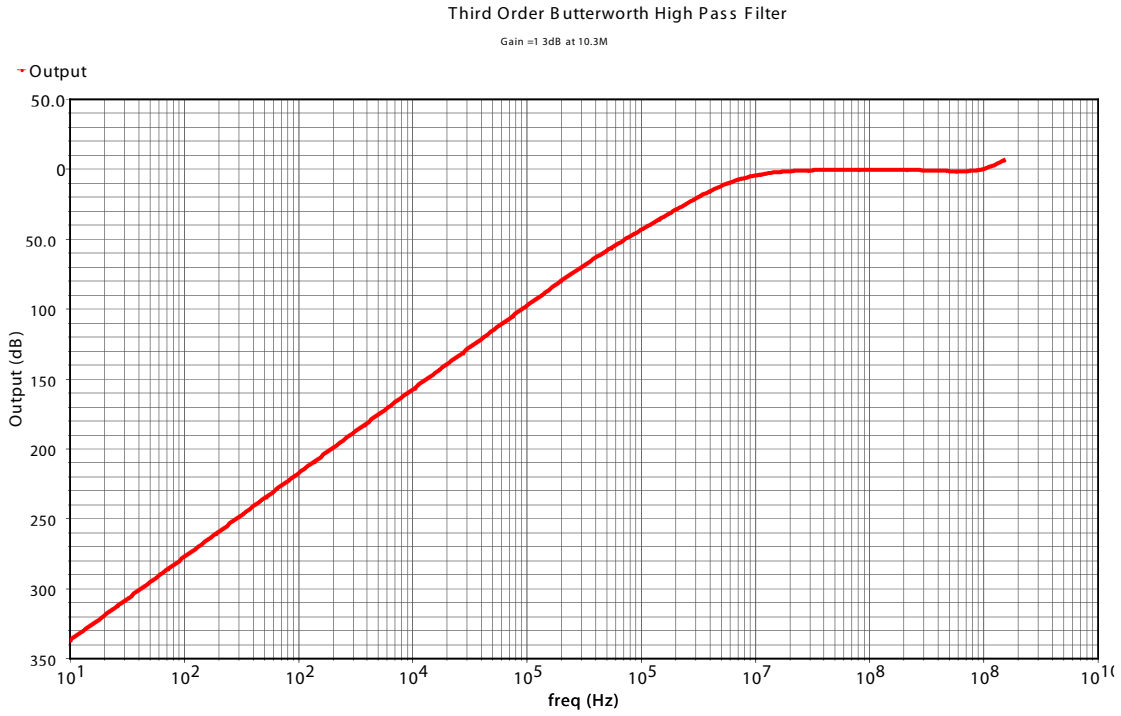
## 7.2 Temperature Measurement

This section describes the application of using FPAA sub-components to implement a bigger system, namely, a temperature monitoring block. The application uses the BGR, the DDA, some passive components and the inter-CAB tracks of the FPAA.

To measure the temperature, we need to find a physical value that has a stable and accurate relationship with temperature and compare it with a temperature independent parameter. Although at a first glance that the PN junction voltage $V_{BE}$ might be an option, that's not a good design because $V_{BE}$ temperature dependence is non-linear and varies significantly from fabrication process to process. As introduced in Chapter 6, the voltage difference between two PN junctions operated at different current densities is an excellent choice. This value has a precise PTAT temperature dependence behavior (equation (6.3) an (6.4)), and can be easily derived from the BGR as shown in figure 7.11. Theoretically, the floating voltage $\Delta V_{BE}$ across $R_1$ can be used differentially, but $Q_2$ collect voltage is at the lower boundary of the DDA common-mode input and may be out of this range at low temperature. So the PTAT voltage was developed across $R_2$. The current $I_s = \Delta V_{BE}/R_1$ is not an




(a)




(b)

Figure 7.10: A third order Butterworth high-pass filter based on Sallen-Key Topology

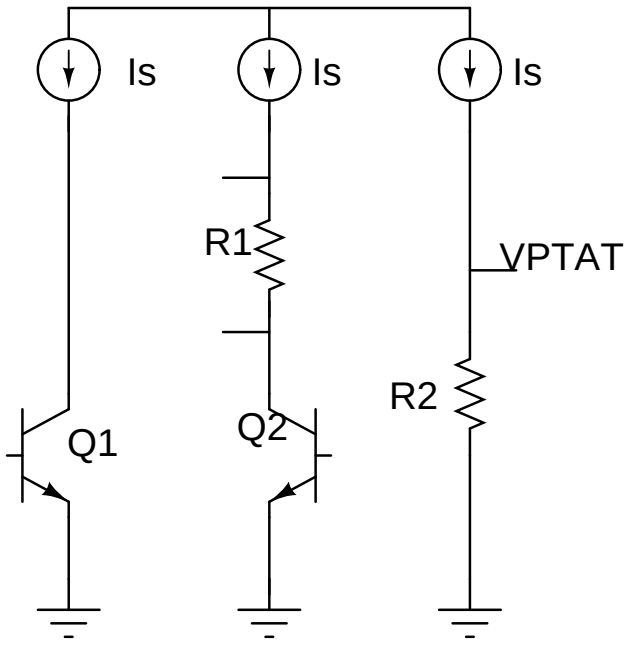


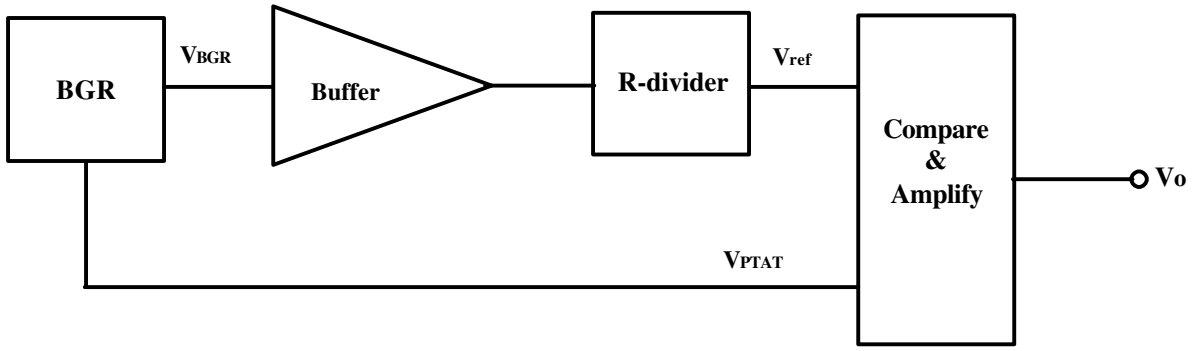

Figure 7.11: A simplified schematic of the generation of $V_{PTAT}$

accurate PTAT value, but $V_{PTAT} = \Delta V_{BE} \cdot (R_2//R_1)$ would be since same type of resistors were used for $R_2$ and $R_2$. By careful layout design, this ration can be kept accurately and is almost independent of temperature.

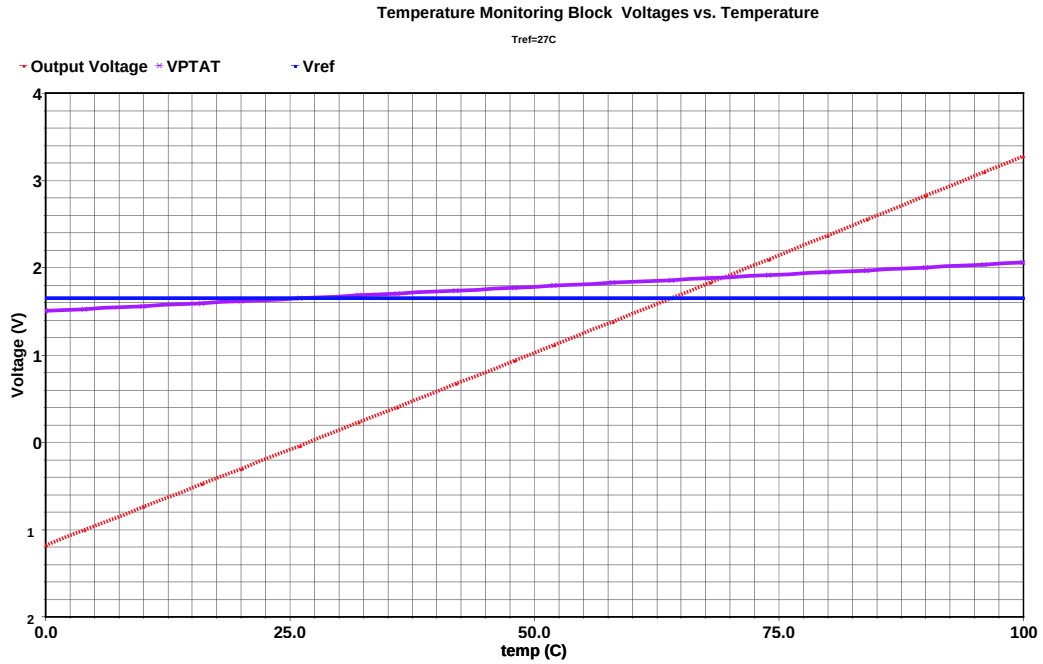
Figure 7.12 (a) shows the overall temperature monitoring block diagram. The output from BGR is a reference that is not capable to drive resistive load, so a buffer amplifier was used. The $V_{ref}$ and $V_{PTAT}$ were pre-calibrated to the same value at room temperature $27°C$, which serves as a reference point. As temperature changes, their difference is compared and amplified (by five in this design) by the DDA. The temperature can be read according to the following formula:

$$T = T_{ref} + \frac{V_T - V_{ref}}{K} \tag{7.8}$$

where $T_{ref}$ is $27°C$, and $K$ is the slop of the output voltage as a function of temperature.




(a)




(b)

Figure 7.12: Temperature monitoring/measurement block (a) diagram; (b) result (0-100°$C$)

When this implementation is combined with some digital circuits and an ADC, the output voltage may be directly converted into temperature reading. A more straightforward but useful application is to use it to monitor the critical temperature. For example, the $V_{PTAT}$ can be pre-calibrated to be a value smaller than $V_{ref}$ until the chip temperature reaches the critical temperature. Thus the output will trigger a positive pulse, which can be used to shut down a certain circuit block or to lower the power.

## 7.3 A Hierarchical Implementation of an 8-bit Two-Step ADC

FPAA is essentially an analog system. All the applications developed previously are still pure analog signal processing. However, using the flexibility provided by laser Makelink, namely, reconfiguration at metallization level, the array based approach can be extended further into a hierarchical design methodology.

Analog-to-Digital converter is probably the most important mixed-signal circuit, which builds a bridge between the real analog world and the digital domain. There are many types of ADC's [97], [100], [101]. Flash ADC has a simple architecture and the fastest speed. Today's 6-bit CMOS flash can be operated at GSPS (Giga sample per second) speed [102], [103], but it has a prominent drawback - the number of comparators grows exponentially with the number of bits. Increasing the quantity of the comparators also increases the area of the circuit, as well as the power consumption. The folding-and-interpolating architecture originally developed for bipolar technology can reduce the number of comparators. But the folding amplifiers are usually open-loop configuration to provide the high frequency operation. The large offset of CMOS implementation makes it difficult to implement open-loop amplifiers. Also, since the coarse stage and the folding stages are inherently different, the timing error is going to be a critical issue [101], [104], [105]. Another option would be pipelined architecture. It significantly reduces the number of comparators. High speed, high resolution and low power may be achieved simultaneously by using this architecture. However, its long latency (for a N-bit pipelined ADC, the latency is usually N clock cycles or longer) may exclude it to be used in many applications

[100]. Thus, a two-step flash architecture was chosen for this 8-bit ADC design.

Figure 7.13 is the block diagram of this two-step ADC. The traditional flash architecture is separated into two subrange flash ADCs with feed-forward circuitry. After the fully differential signal is sampled, a coarse estimate of the input signal is obtained by the by the most-significant-bit (MSB) ADC, or coarse ADC. The result is then converted back to an analog voltage with the DAC and subtracted from original input. The residue from the subtraction is multiplied by $2^4$ and fed into the fine ADC (LSB ADC) to generate the final four bits. The coarse conversion, DAC conversion, subtraction and fine conversion have to be completed in the sampling period. Among them, the subtractor is the slowest part. Because the major error would be from the MSB ADC and the error in fine ADC is at LSB level, to improve the speed, the residue in this design was multiplied by 2 instead of 16. Comparing to 8-b flash which requires 255 comparators, this two-step architecture only needs 30 comparators. Most of the analog components in this design were based the DDA and CAB structure.

**Sample and Hold** The front-end S/H circuit plays a crucial role in the performance of the two-step flash ADCs. Without the S/H circuit, the maximum allowable slew rate of the input signal is severely limited. This occurs because if the analog input signal varies rapidly in the conversion period, then the signal level digitized by the DAC is not equal to the signal sensed by the subtractor. To avoid the errors during the conversion period, a high speed amplifier with large slew rate is desired. The DDA developed in Chapter 5 is an excellent choice.

Typically, the switches in the ADC are implemented with MOS transistors.

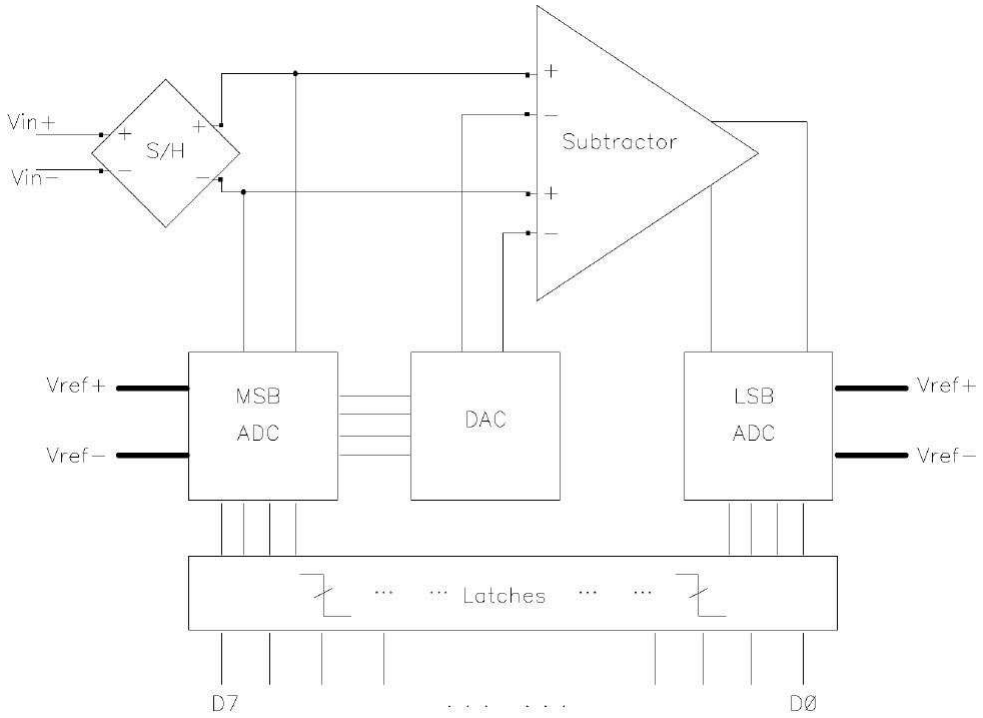
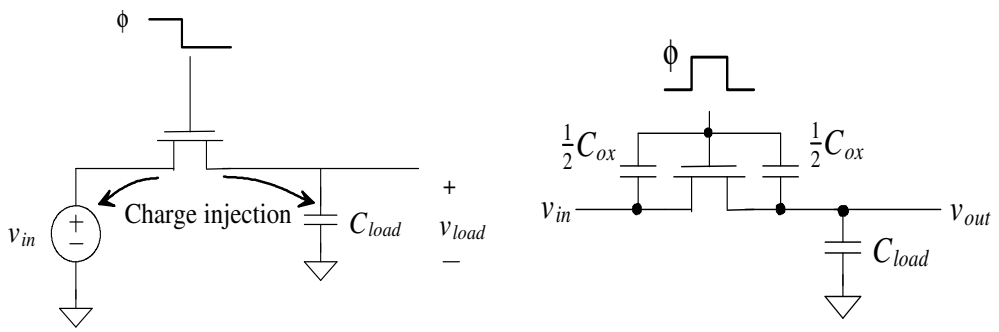



Figure 7.13: A Fully Differential 2-step Flash ADC Diagram

It is important that the MOS switches have a linear transfer function and constant on resistance, which is effectively independent of input voltage so the RC time constant for charging the capacitor is constant for all input signal amplitudes. More importantly, two classical non-ideal effects associated with MOS switches usually limit the performance of these switches. These two effects are known as charge injection and clock feed-through [106], [107].

When a MOS switch is on, it operates in the triode region and its drain-to-source voltage, $V_{DS}$, is near zero. During the time when the transistor is on, it holds mobile charges in its channel. Once the transistor is turned off, these mobile charges must flow out from the channel region and into the drain and the source. This is charge injection. Because the amount of charge in the channel is signal dependent,



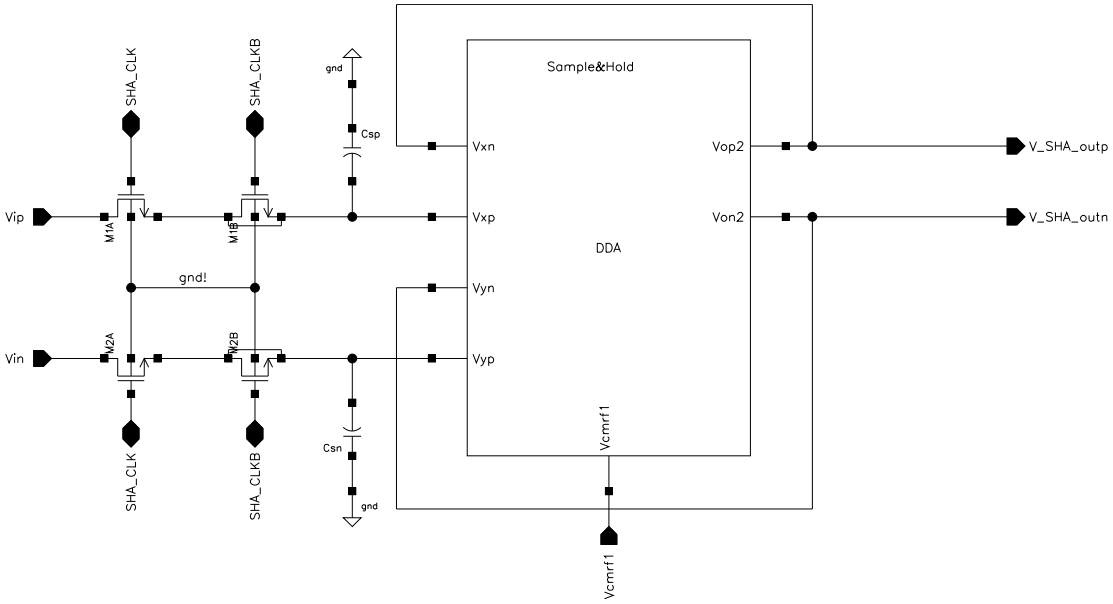

(a) (b)

Figure 7.14: (a) charge injection; (b) clock feedthrough

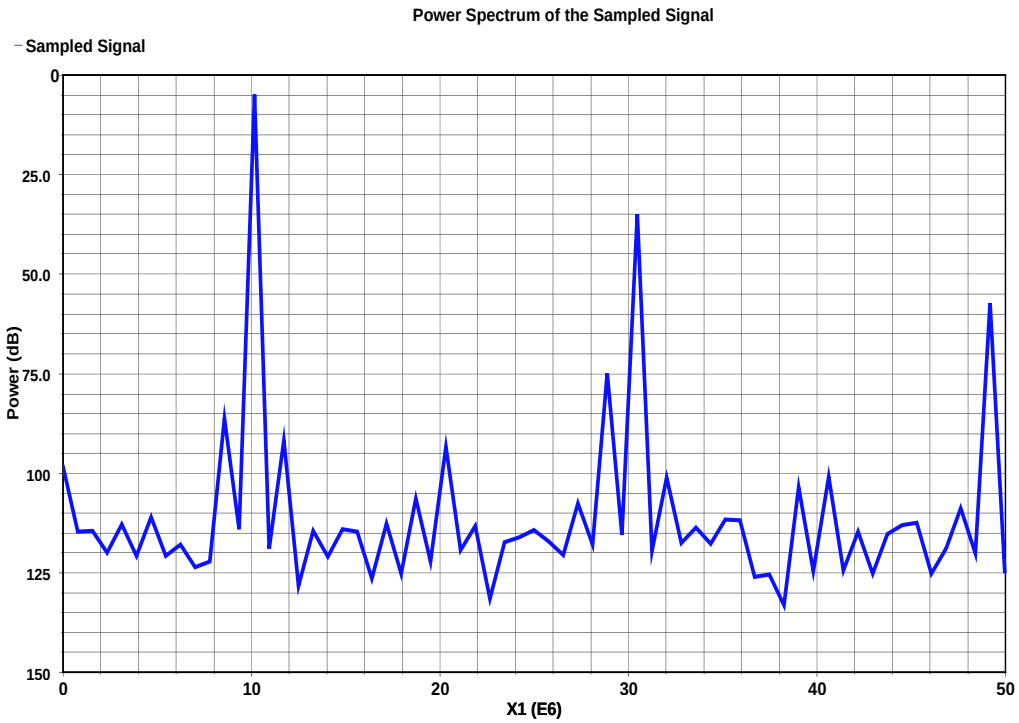
this charge injection error is non-linear and difficult to be removed completely. The clock feedthrough is due to the MOS capacitance. It can be largely removed by fully differential operation.

Figure 7.15 (a) is a simple CAB/DDA based implementation. A dummy NMOS transistor in serial with the main switch has 1/2 the $W/L$ of the main switch transistor. To the first order, the injected charges are absorbed by the dummy switch [102]. However, from the power spectrum of the sampled signal ($100MSPS$ with $10.1258MHz$ input signal), the THD (total harmonic distortion) is smaller than $37dB$. Even without considering the noise, this parameter itself already limits ADC's ENOB (effective number of bits) smaller than 6 bits. So this topology is inadequate for close to hundred MSPS operation of an 8-bit ADC.

In this design, a fully differential bottom plate S/H circuit was chosen [108], as shown in figure 7.16 (a) and (b). At time $t_1$, the CLK1 MOS switches turn off. The charge injection and clock feed through resulting from this action are common-mode signal and can be largely reduced with the fully differential topology.




(a)




(b)

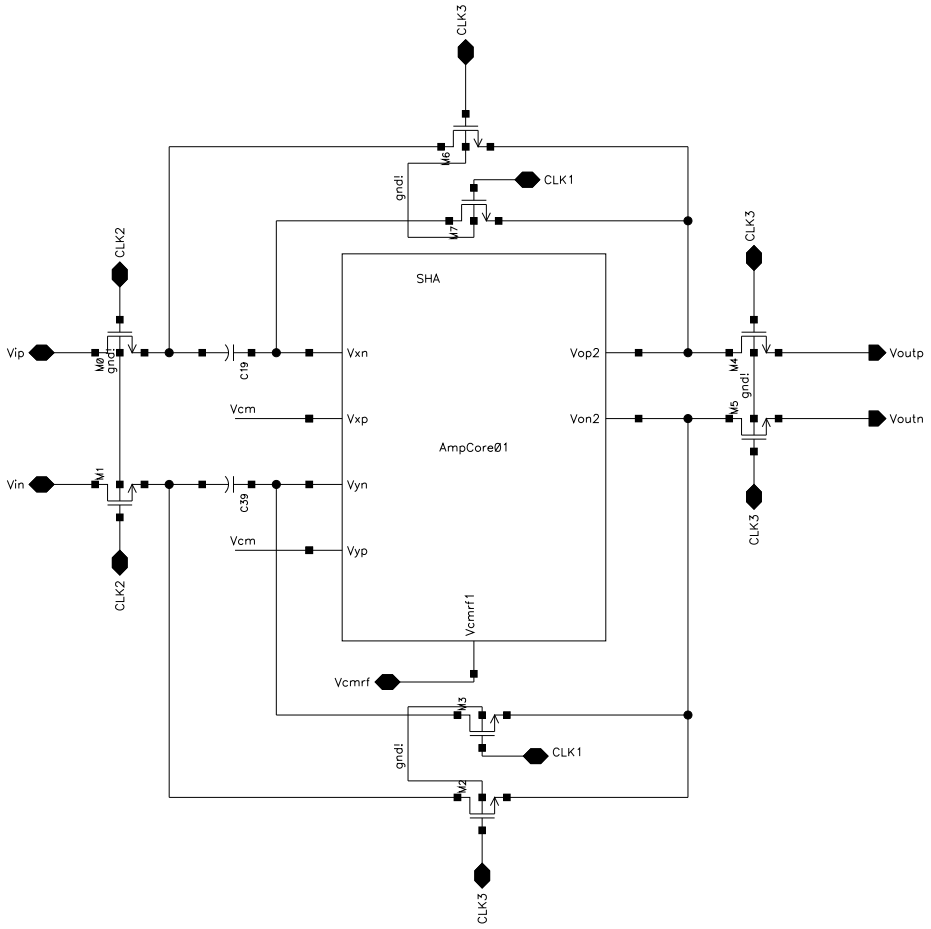
Figure 7.15: A fully differential S/H based on DDA follower (a) schematic; (b) power spectrum of the sampled signal

Attention should be paid that the time internal between time $t_1$ and $t_2$ should be small because the DDA is in open-loop operation. When CLK2 controlled switches turn off, due to the high impedance at DDA's inputs, which makes the sampling capacitors' top plate floating, the sampled voltage won't change. At last, the charge injection and clock feedthrough due to CLK3 off can also be reduced differentially. The bottom plate of the integrated capacitor is always associated with large parasitic capacitance. The BPS connection brings two advantages: (1) the substrate coupling noise doesn't directly feed into the DDA, nor does it affect the sampled voltage; (2) reduced gain error [97].

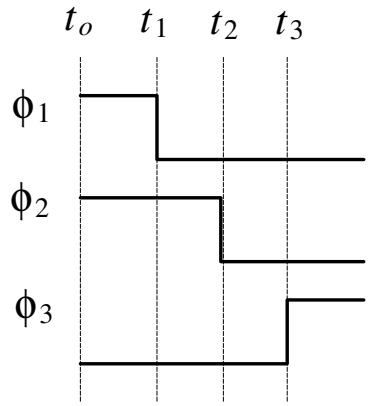
Figure 7. 17 shows the power spectrum of the sampled signal. With this topology, the THD is improved to near $60dB$, which is sufficient for 8-bit ADC.

**Comparator** The speed of the ADC strongly depends on the comparator design. The comparison process is effectively a binary phenomenon that generates the logic HIGH or LOW. The op amp may be used directly as a comparator, but its comparison speed is often very slow. Even in open-loop configuration, the time required to settle the valid logic output is still not tolerable for high speed ADC's. Since the comparator needs not to be linear or closed-loop, positive feedback can be introduced to attain near infinite gain, at the mean time, improve the speed. Attention should be made that, to avoid unwanted latch-up, the positive feedback must be enabled only at a proper time. This usually means the comparator gain changes from a small value to a very large value at proper timing [101].

In order to use as much existing components in the FPAA as possible, the following design was developed. The comparator consists of a preAmp and feed-




(a)




(b)

Figure 7.16: A fully differential BPS S/H (a) schematic; (b) timing graph

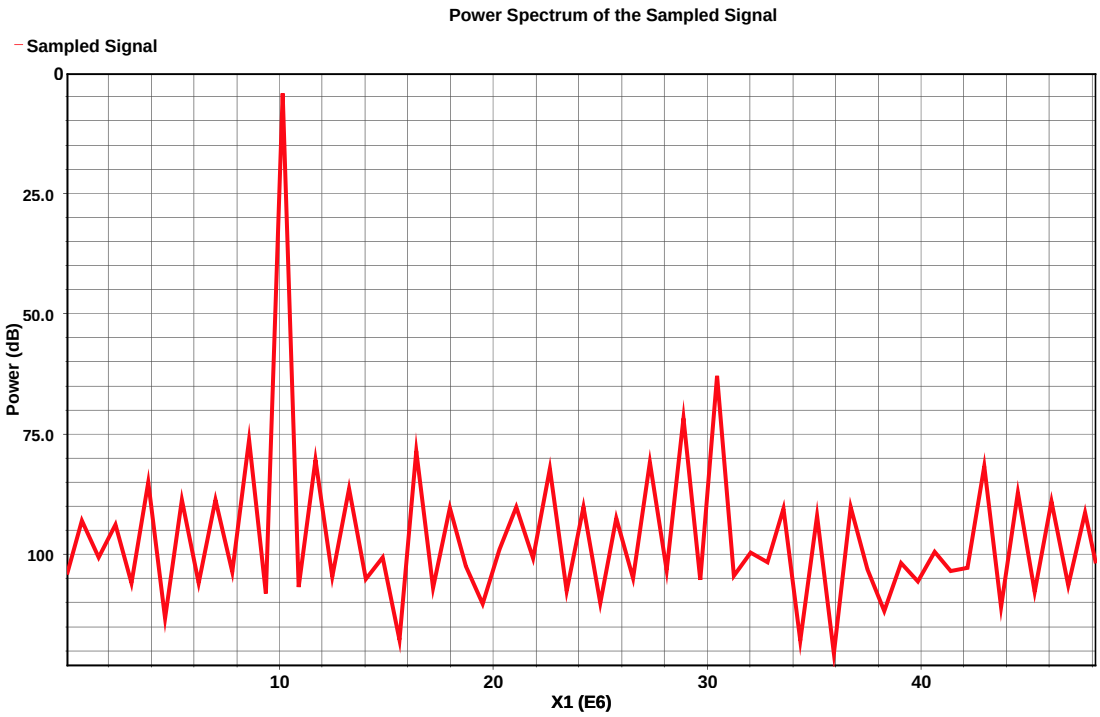
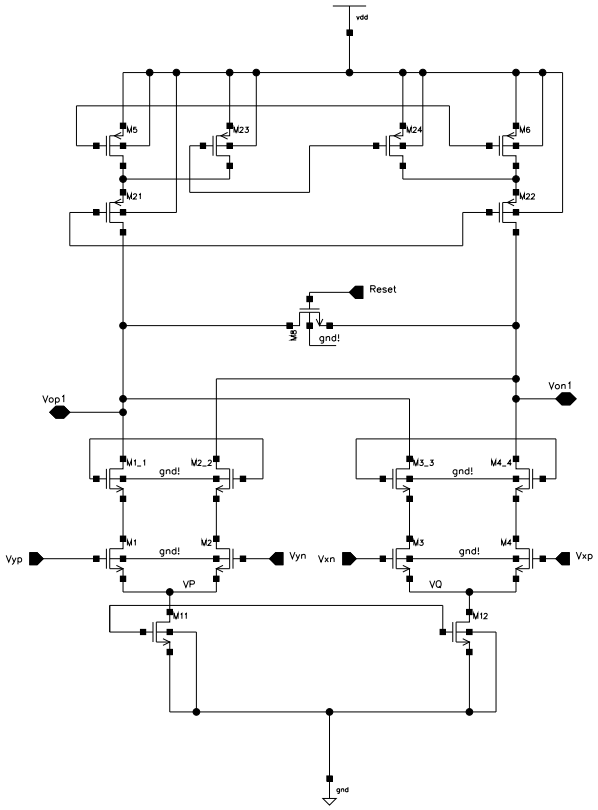


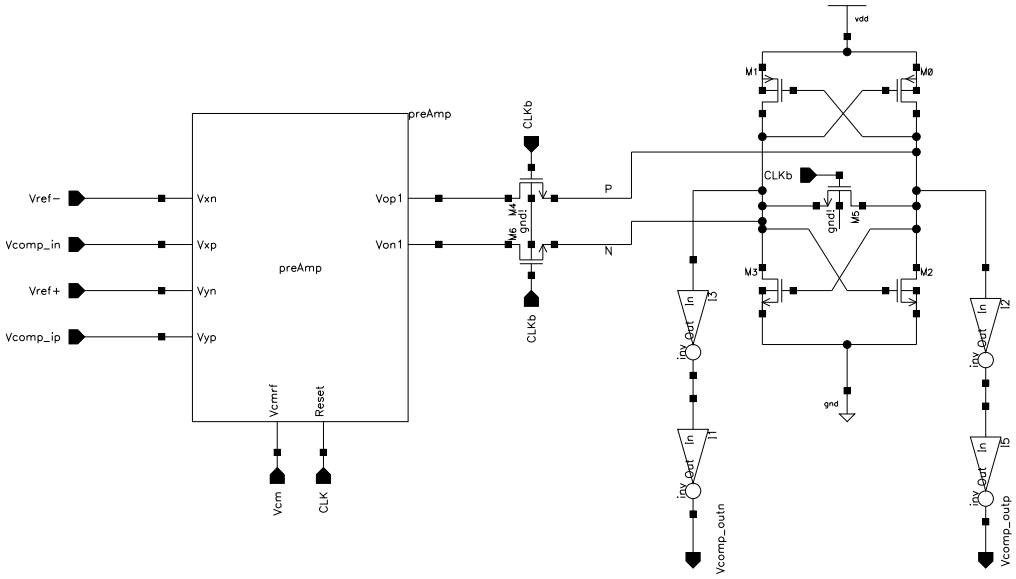

Figure 7.17: BPS: Power spectrum of the sampled signal

forward latch. The preAmp is actually just the first stage of the DDA with a reset switch connecting the two outputs. It amplifies the input difference by a small amount so as to cover the offset of the latch, which is difficult to cancel. It also functions as a buffer between the resistor ladder and the latch. An added benefit of using this structure is, the cascoding devices help shielding the kick-back from the regenerative outputs of the latch, thus reduce the ADC bit-error-rate (BER) due to the fluctuation in the resistor ladder caused by the kick-back noise. To complete the comparator design, an extra block, feedforward latch, which doesn't exist in the pure analog array has to be added. Through positive feedback, the latch will generate the final logic level quickly. Two sized inverters were added as logic buffer.

At sampling clock high (CLK = 1), the preAmp is in reset mode which removes any residue left in the previous comparison process. During the hold mode




(a) preAmp




(b) Comparator

Figure 7.18: The DDA based comparator

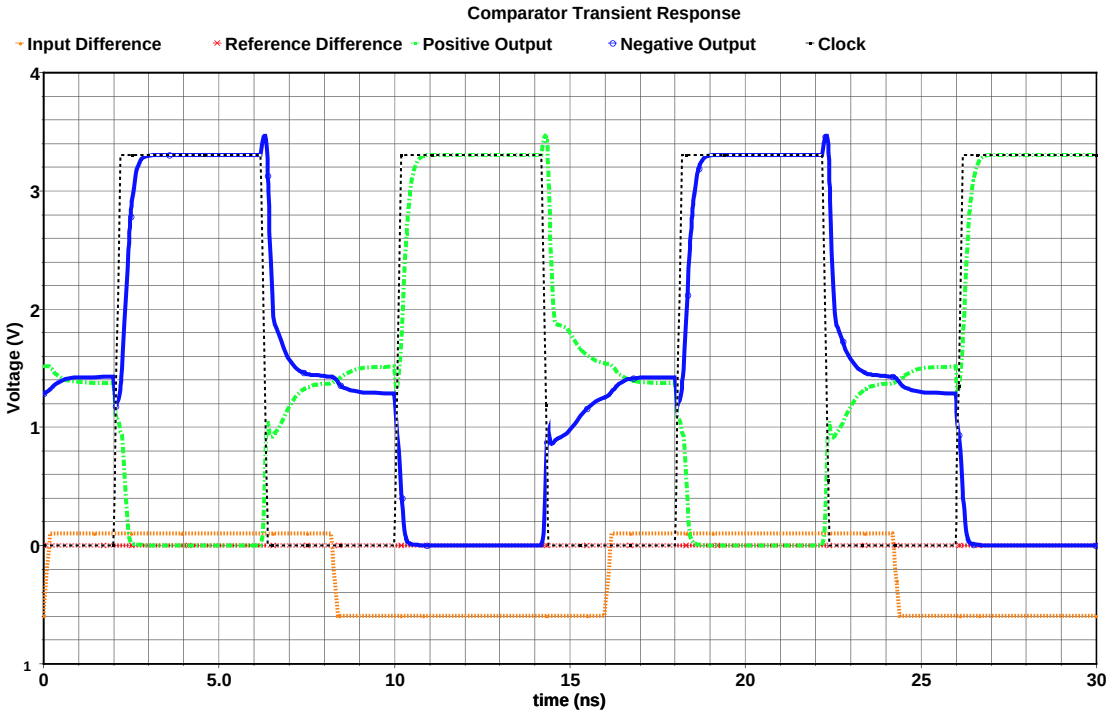



Figure 7.19: The DDA based implementation of the subtractor

(CLKb=1), the outputs of the preAmp are short to the latch thus forming a low gain but high speed amplifier (pre-amp the input difference). For the latch, it regenerate the logic outputs while preAmp is in the reset mode. Figure 7.19 shows the two reset switches can significantly reduce the "over-drive" recovery time of the comparator. The propagation delay is about $0.5ns$ from negative full scale difference to positive $1LSB$ difference.

**Subtractor** Using DDA and some CAB components, the implementation of the subtractor is simple. It only takes one amplifier plus several feedback resistors, as shown in figure 7.19. Since gain directly trade bandwidth, the subtractor has been the slowest component of this design. To somewhat increase the speed a bit, a gain of 2 instead of 16 was used. Of course, this reduces the value of the LSB in the fine ADC. This puts more stringent requirement on the comparator offset, which

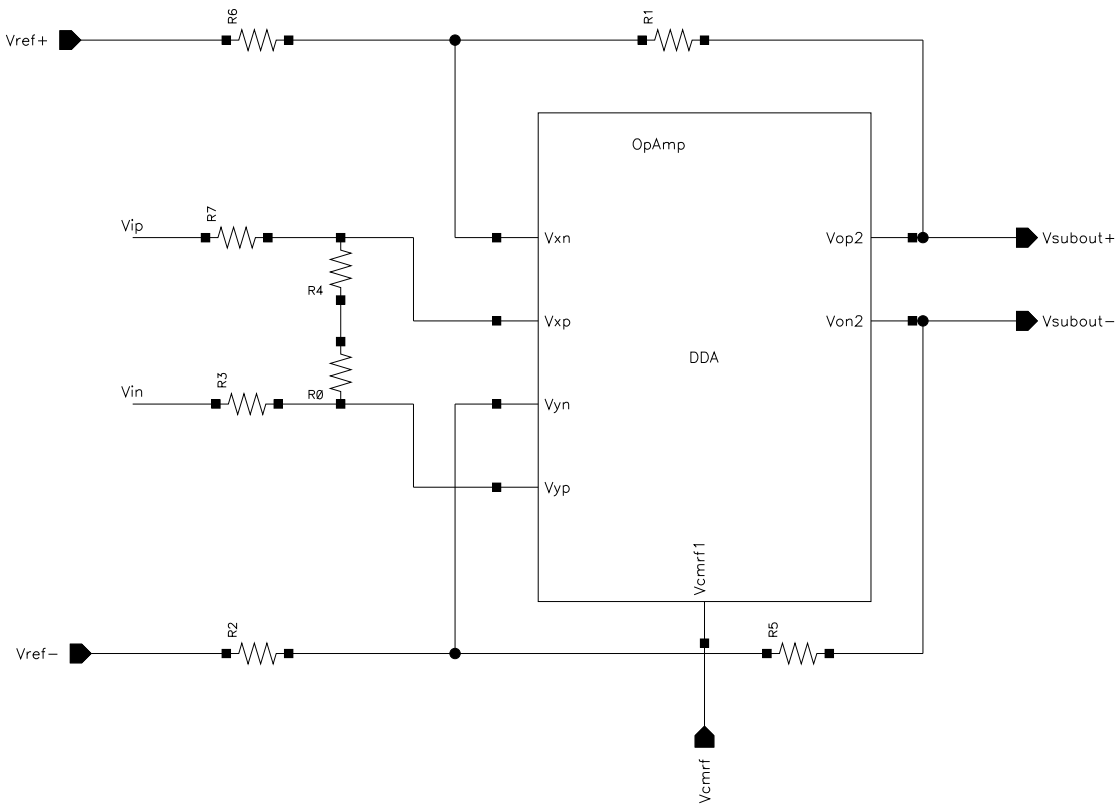



Figure 7.20: The DDA based implementation of the subtractor

may be reduced by using laser Makelink trimming.

Other ADC components employ the existing designs in the Analog System Design Lab. To reduce the switch parasitics, a folded resistor string DAC was used. The encoder is a gate level implementation which contains the bubble error correction scheme [109]. The overall two-step ADC achieves speed of $70MSPS$, consumes $154mW$ and occupies $1600x1600um^2$. Comparing to the similar full custom design, it is about 50% slower and takes more space. Although this is not an optimal design, it demonstrates the flexibility of the laser Makelink based hierarchical design approach. And the overall the design cycle was significantly reduced. Therefore, when the short turn-around time is the primary concern, this design methodology is going to be an attractive option.

# Chapter 8

# Conclusions and Future Work

The main contributions of this work can be summarized as follows: (1) **CAD** proposed a generic arrayed based FPAA architecture and a flexible CAB topology; improved a pathfinder negotiated routing algorithm and implemented the algorithm in C for a prototype FPAA. investigated and proposed analog constraints for performance-driven routing. (2) **Analog Sub-System Design** invented and designed a novel differential difference op amp; designed two bandgap reference circuits including a low voltage version; based on the prototype FPAA architecture, developed several application examples; (3) **Laser Makelink** studied and designed various laser Makelink test structures on different CMOS processes and BiCMOS copper process; invented a novel offset trimming method using laser Makelink; proposed some preliminary ideas on laser Makelink reconfiguration on analog circuits.

However, the FPAA development is a very complex project that requires a significant amount of work in CAD, architecture and circuit design. Also, although the idea of laser Makelink reconfiguration was proposed, its application in many analog circuits have not been fully investigated and experimentally verified. Thus, the following work is expected to be continued in the future.

(a) **Analog CAD Design Methodology** Due to the fundamental differ-

ence between analog and digital system, the full automatic design synthesis can not be obtained for analog IC design, but we may still borrow some digital IC design methodologies. Instead of a traditional bottom-up design, a top-down hierarchical process can be employed. Design entry can start from Verilog-A or AHDL (analog hardware description language) block. The overall system performance can be estimated at the early design stage thus preventing the risk of insufficient design or over-design. Then the HDL-based design can converted or optimized with supporting IPmodule library.

(b) **Analog IPmodule Library Development** Besides the universal op amp unit and the accurate reference blocks, other analog building blocks also need to be developed. For examples, a fully differential wideband buffer with rail-to-rail input and output range is desired. To properly drive the off-chip load, a high speed I/O block needs to be developed. At a higher design level, various application specific circuit functions should be added into the IPmodule library as the pre-qualified design for the end users. This includes filters, control circuits etc.

(c)**Applications** Although some of the FPAA functionalities have been demonstrated, the exact, practical application examples are still not clear. More investigation needs to carried on for field application.

(d) **Laser Reconfiguration** Laser Makelink is an powerful programming technology that provide tremendous design flexibility. It can used as a trimming method, and further more, to reconfigure the analog circuits into different forms and to modify them to satisfy different application needs. For example, the active filter developed in the previous chapter suffer from the poor precision of the integrated passive

components thus inaccurate filter response. And it is difficult to tune. OTA-C (or called $G_m$-C) would be an excellent choice for continuous-time filter implementation, because it provides the flexibility of "tuning" by adjusting the transconductance. Although the DDA op amp can also be used as an "OTA", but due to its two stage structure, the second pole is very close to the dominant pole. So its speed is still limited. Using laser Makelink, we can "cut-off" the second output stage, and just use the first stage as an OTA. This way FPAA can be "reconfigured" for filter type of applications. The above method effectively gives us two critical building blocks: a high-gain, flexible op amp and a high speed OTA. Or we can still keep the two stage structure with the original class AB output stage to provide the necessary swig, but reconfigure the first stage as a diode connected differential input stage. The application of laser Makelink reconfigurability actually has been beyond the original FPAA design concept. This can be treated as a hierarchical design approach that's applicable to SoC's or other Mixed-Signal ASICs.

Appendix A

Chip Layout

## A.1 Laser Makelink Test Chips

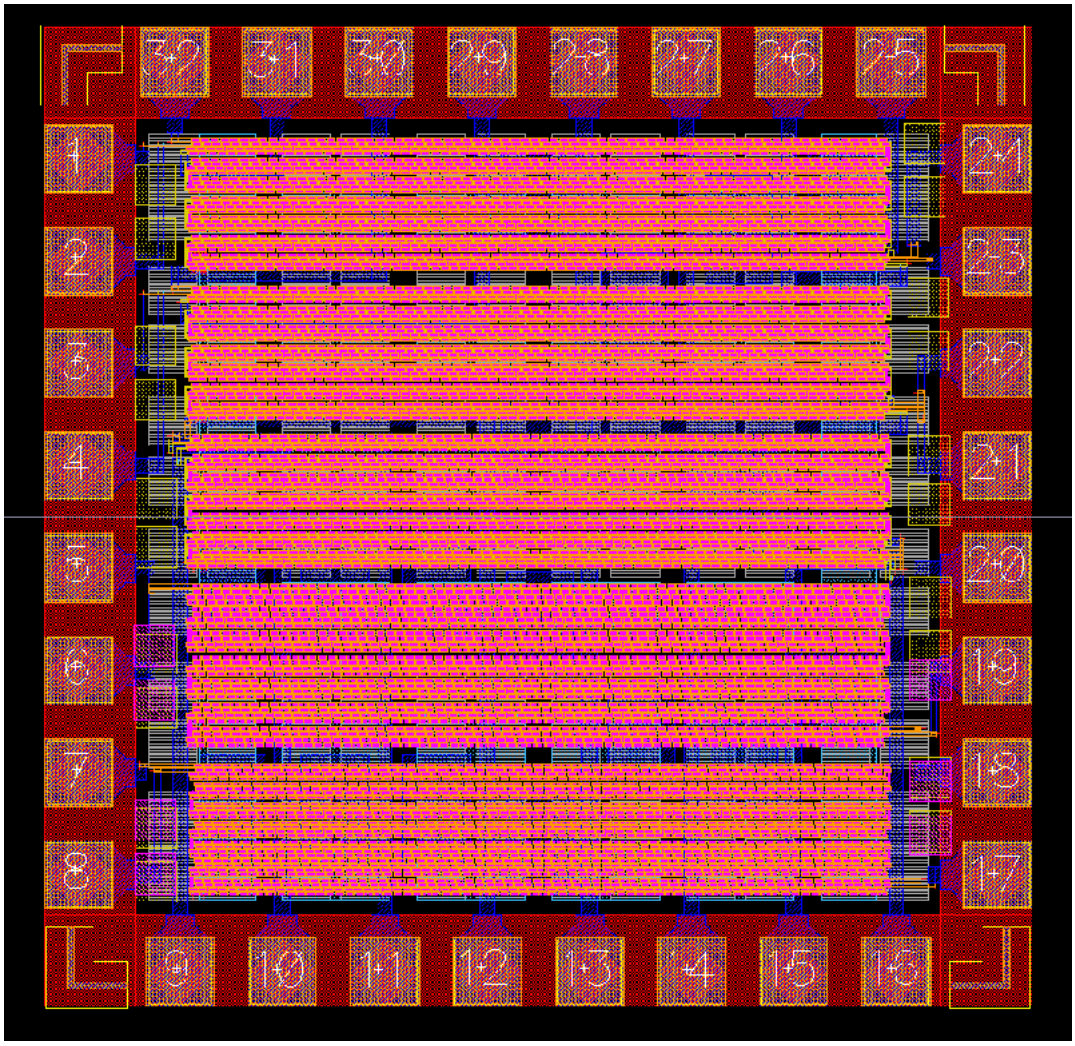



Figure A.1: Al Makelink test chip - NSC 0.18um CMOS

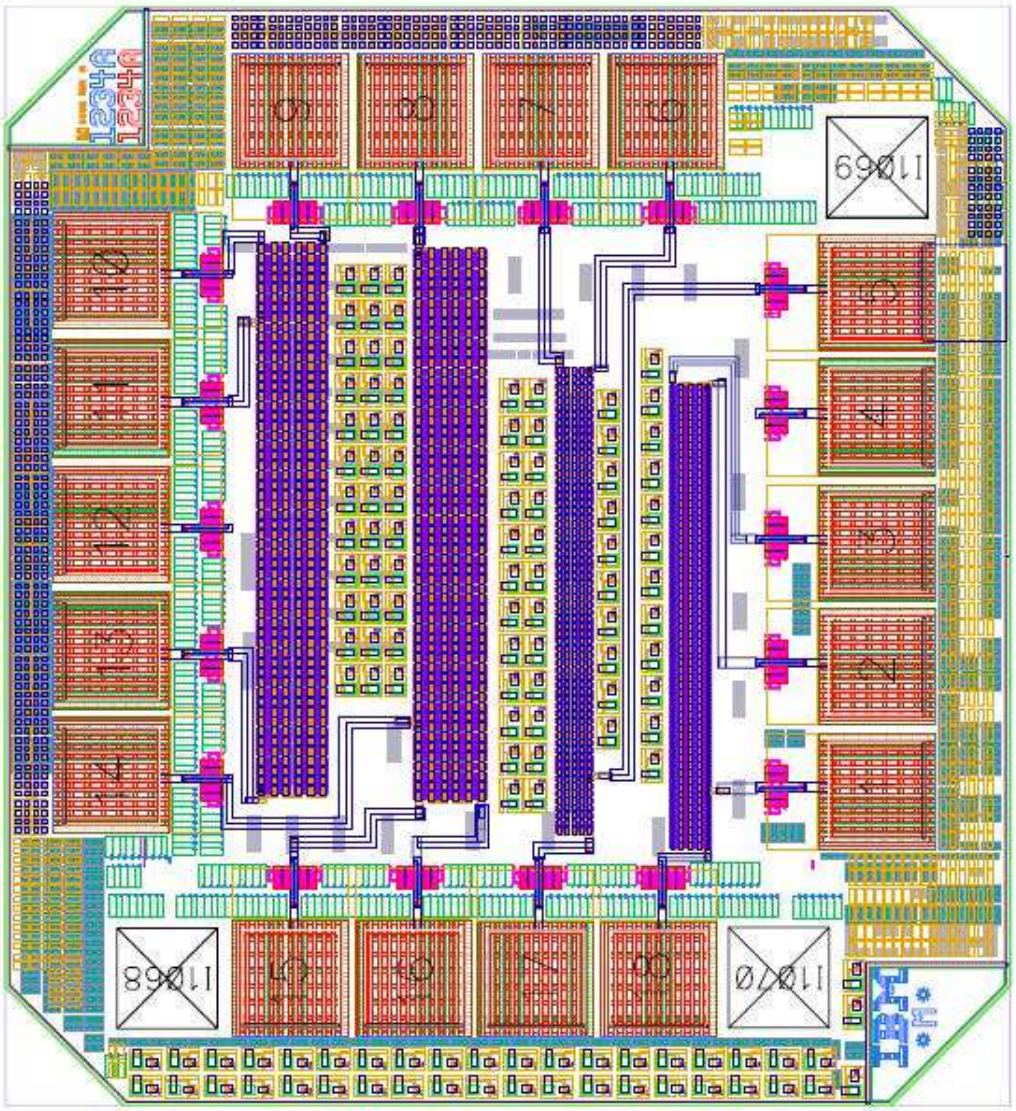


Figure A.2: Cu Makelink test chip - IBM 8HP 0.13um BiCMOS SiGe

## A.2 The Fully Differential Difference Amplifier

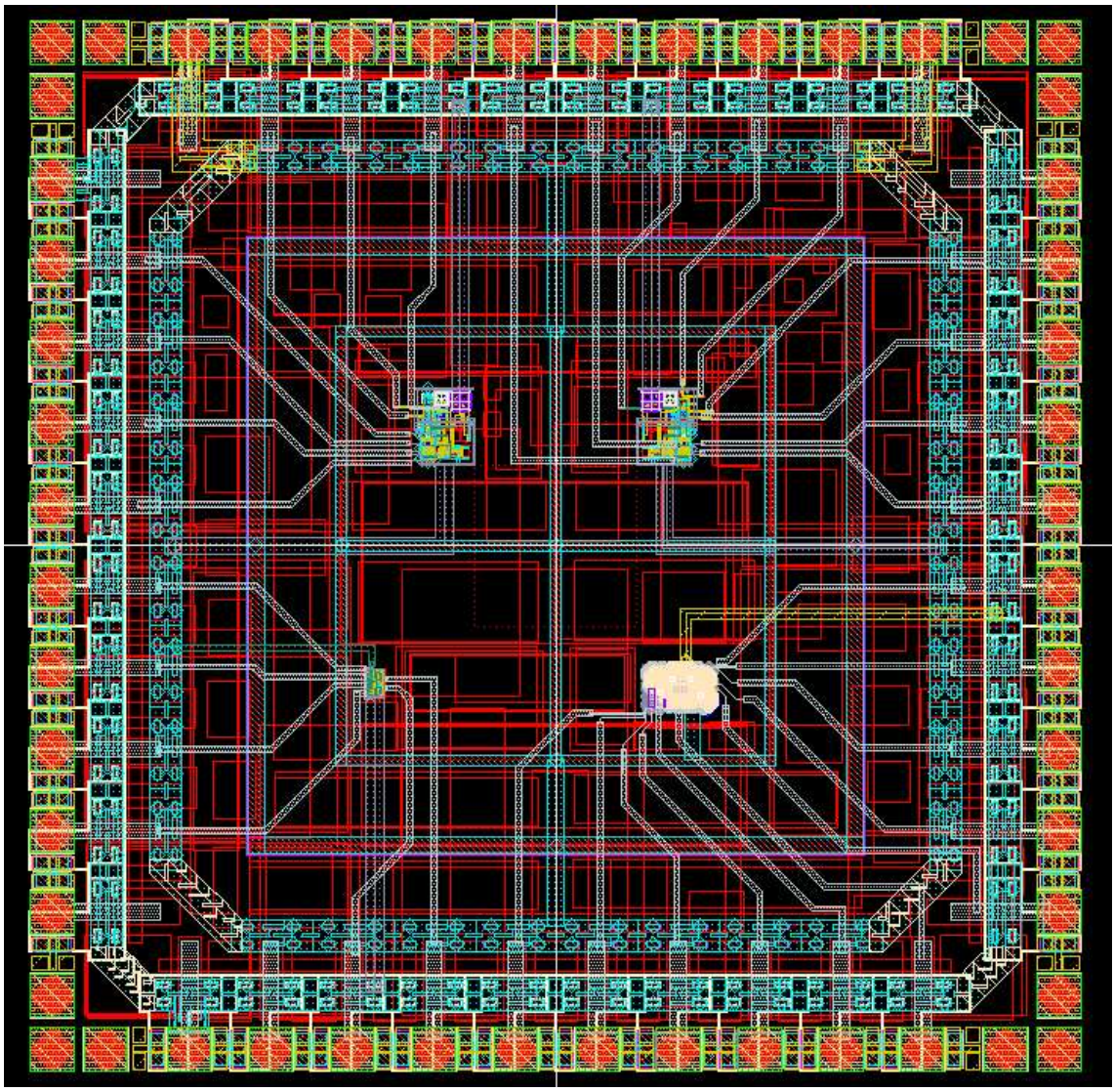


Figure A.3: The fully differential difference op amp - TSMC018 CM process

## A.3 The Bandgap Reference

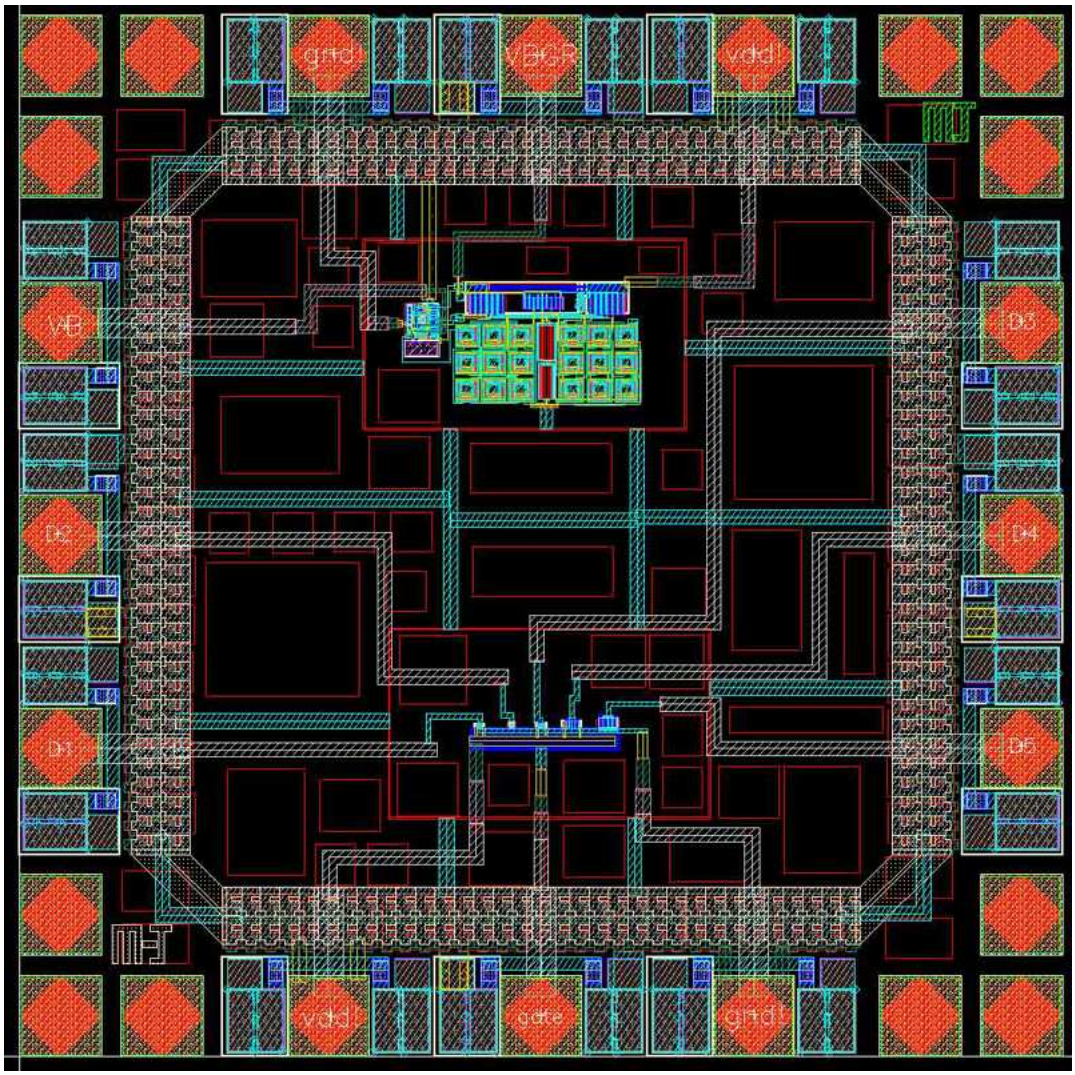



Figure A.4: The first order bandgap reference chip with test transistors - TSMC018 CM process

## A.4 Two-Step ADC

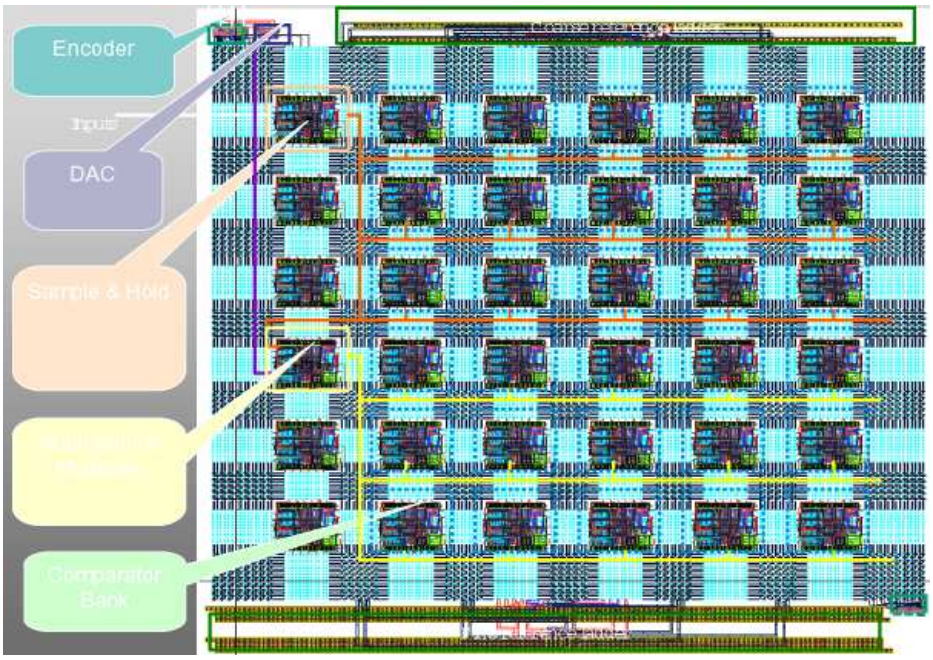



Figure A.5: The two-step flash analog-to-digital data converter - TSMC018 CM process

# Appendix B

# FPAA Router Documentation

# I. Variables

- ***s_net (t_net):*** defined in header file netlist.h. This is a struct type data structure used to maintain a net. The definition is shown as below:

```
*****************************************************************
struct s_net {
        int index; /* index of this net in the netlist */
        int num_terminals; /* totoal number of terminals of this net */
        int *pTerminals; /* used to maintain the linked list of terminals */
};
typedef struct s_net t_net;
*****************************************************************
```

- ***s_Rtvertex (t_Rtvertex):*** defined in header file route.h. This struct defines a data structure used by the vertices in routing tree RT.

```
*****************************************************************
struct s_RTvertex { int index; /* index: the index of this vertex; */
                    short PQflag;
                    struct s_RTvertex pNext; /* pNext: pointer to the next vertex */
};
typedef struct s_RTvertex t_RTvertex;
*****************************************************************
```

- ***s_RT (t_RT):*** defined in header file route.h. This is a struct type data structure used to maintain a routing tree RT, which is a linked list. Each routing tree RT is corresponding to a partial or complete net.

```
*****************************************************************
struct s_RT {
         int neti; /* index of this RT/net */
        int num_v; /* total number of vertices in the final RT */
        t_RTvertex pRTvertex; /* Pointer to a member of this RT. In this version, it always  */
                              /* points to the head of this routing tree because new vertex  */
                              /* is alway added in the front of the old ones  */
};
typedef struct s_RT t_RT;
*****************************************************************
```

- ***s_PQvertex (t_PQvertex):*** defined in header file pqueue.h. This data structure used to maintain a vertex in priority queue.

```
*****************************************************************
struct s_PQvertex {
                    int index; /*  the index of the vertex in RRG */
                    struct s_PQvertex pParent;  /* the parent of this vertex in RRG, NOT in the
                            context of PQ. This parameter is used in back-tracing stage. */
                    double pathcost; /* the path cost of from current partial routing tree to this
                                        vertex including the cost of this vertex itself*/
};
*****************************************************************
```

- ***pqueue:*** defined in header file pqueue.h. This data structure used to maintain a priority queue.

```
*****************************************************************************
struct pqueue {
        int size; /* number of elements this priority queue actually contains */
        int size0; /* how many vertices in PQ (including the vertices which have been removed */
        int avail; /* the maximum number of elements the array can still hold*/
        int avail0; /* the maximum number of elements the redundant array can still hold */
        int step; /* the number of additional elements to be allocated */
        PQvertex *d0; /* pointer to the starting position of the redundant priority queue */
        PQvertex *d; /* pointer to the current location of the priority queue */
};
*****************************************************************************
```

- ***t_rr_type:*** defined in header file rrgtypes.h. This data structure defines the available types of routing resource vertex.

```
typedef enum {IPAD, OPAD, IPIN, OPIN, CHANX, CHANY} t_rr_type;
```

- ***s_edge_list (t_edge_list):*** defined in header file rrgtypes.h. This data structure used to maintain all the neighbors of a vertex.

```
*****************************************************************************
struct s_edge_list{ int index;
                    struct s_edge_list *pNext;
};
typedef struct s_edge_list t_edge_list;
*****************************************************************************
```

- ***eList:*** defined in header file rrgtypes.h. This data structure maintains the linked list of each vertex's edges.

```
*********************************************************************
typedef struct { int index;
        int size;
        t_edge_list *p;
} eList;
*********************************************************************
```

- ***t_rr_vertex:*** defined in header file rrgtypes.h. This is the main data structure used to describe a routing resource graph vertex.

```
*********************************************************************
typedef struct {
        int index;
        short x;
        short y;
        short ppt_num;
        t_rr_type type;
        int occupancy;
        int capacity;
        int num_edges;
        int *edge_list;
} t_rr_vertex;
/*****************************************************************************
 * index: index of the vertex
```

```
* x, y:  integer coordinates
* type:  What is this routing resource?
* occupancy: how many nets are using this vertex now?
* capacity: how many nets can legally use this vertex?
* ppt_num:  Pin, track or pad number, depending on rr_vertex type.
* num_edges:  number of edges exiting this vertex, i.e. the number of vertexs to which it connects
* edge_list: pointer to the linked list of all its neighbors                    *
*****************************************************************************/
```

# II. Function Descriptions

- ***t_net * read_netlist (char * fname):*** read in the netlist file to be routed. Return a t_net type pointer, which points to the start position of the netlist stored in memory.

- ***void free_netlist (t_net * net):*** free the memory occupied by the nets to be routed.

- ***struct pqueue * pqinit (struct pqueue *, int):*** initialize priority queue. Return a pqueue type pointer to the start position of the priority queue in memory if success. If memory is insufficient, return a NULL value.

- ***int pqinsert (struct pqueue *, PQvertex):*** insert an element into priority queue. Return 1 if the insertion successes. Return 0 if insertion fails.

- ***PQvertex pqremove (struct pqueue *):*** remove the highest-ranking element from the priority queue. Return a pointer to the memory location of that element. Return a NULL value if the removal fails.

- ***int print_PQ (struct pqueue *q):*** print out the content of the priority queue. Return 1 if it successfully prints the priority queue. Return 0 if the priority queue is empty.

- ***void free_PQ (void):*** free the memory occupied by priority queue.

- ***void init_RT(int neti):*** initialize a routing tree (RT) for each net.

- ***void add_v_RT (int vindex, int neti):*** add a vertex with index "vindex" into the corresponding routing tree RT.

- ***int is_in_RT (int jv, int neti):*** check if a sink is already in routing tree. Return 1 if this sink is already contained in routing tree. Return 0 if this sink hasn't been added into routing tree.

- ***void route (int neti):*** the primary subroutine used to route a single net.

- ***void print_RT (int neti):*** output the vertices currently contained in routing tree. This subroutine is for debugging purpose.

- ***void update_p (int vindex, int trend):*** update vertex's present congestion cost and total cost.

- ***void init_PQ_to_RT (int neti, int * reached):*** initialize priority queue with the vertices currently resided in routing tree.

- ***void enqueue_PQ (int jn, PQvertex pParent):*** add the fanout verteice of vertex m (which is removed from PQ) to priority queue and calculate the pathcost.

- ***int is_sink (int jsink, int neti):*** justify if this is a sink of net i. Return 1 if it's a sink. Return 0 if it's not a sink.

- ***void build_rrg (void):*** build routing resource graph and dump it out into a RRG file.

- ***void add_***_pads (int i, int j):*** add pad at position (i, j) into the routing graph.

- ***void add_cab_pin (int i, int j):*** add pins of the CAB (i, j) into the routing graph.

- ***void add_tracks (int i):*** connect tracks in channel x and channel y if there's a switch box

- ***void creat_edge_list (int present, int neighbor):*** add neighbors of current vertex into the linked list and count the total number of edges.

- ***void free_rrg (void):*** free the memory occupied by the routing resource graph

- ***int get_vertex_index (int i, int j, enum t_rr_type rr_type, int ppt_num):*** calculate the vertex index at specified position. Return this vertex's index number.

- ***void init_cost (void):*** initialize the cost functions for all vertices.

- ***void free_cost_mem (void):*** free the memory occupied by the cost functions.

- ***void update_h (int vindex):*** update history congestion function.

- ***void output_RT (int neti):*** print out the vertices in routing to screen for debugging purpose.

- ***int overuse (void):*** check if overuse exists. Return 1 if overuse exists. Return 0 if no overuse.

- ***double dump_RT (char *fname):*** dump out the finished routing into a file. Return the track usage rate for this routing.

- ***void free_RT (void):*** free the memory occupied by routing trees.

- ***void * my_malloc (size_t size):*** allocate a block of memory. Exit program if no sufficient memory.

- ***int odd_or_even (int dividend, int divisor);*** justify parity of an integer. Return 1 if the integer is an odd number. Return 0 if the integer is an even number.

# III. Instructions

The router program was developed on Windows platform with Microsoft Visual C/C++6.0. The executable can by built by: start Microsoft Visual Studio, create a new workspace and a new project, add all the source files into this project and then build. Or simply copy all the files from 1 – 3 into a directory and click build button in Microsoft Visual Studio.

***Usage***: name_of_executable input_netlist_file output_file

Where input_netlist_file is the netlist to be routed, output_file is user specified output file to store finished routing result. For example, in a command line, type:

pathfinder test.net test.r

- **File List**

  1. **Source Files:** *main.c, netlist.c, pqueue.c, route.c, rrg.c, utils.c*
  2. **Header Files:** *netlist.h, pqueue.h, route.h, rrg_funcs.h, rrg_types.h, utils.h*
  3. **Workspace File & Project Files:** *pathfinder.dsw, pathfinder.dsp*
  4. **Sample input netlist file:** *test.net*
  5. **Executable:** *pathfinder.exe*
  6. **Generated routing resource graph file:** *rr_graph.out*
  7. **Sample output file:** *test.r*

# BIBLIOGRAPHY


[1] http://focus.ti.com/docs/pr/pressrelease.jhtml?prelId=sc04074

[2] http://www.st.com/stonline/press/news/year2004/t1573h.htm

[3] http://www.eetimes.com

[4] Databeans Inc.(http://www.databeans.net), "2005 Analog Markets Worldwide".

[5] Microelectronics Design Center, *Simplified Digital Design Flow*, Swiss Federal Institute of Technology.

[6] Stephen M. Trimberger et al., *Field Programmable Gate Array Technology*, Kluwer Academic Publishers, 1994

[7] http://www.anadigm.com

[8] M. Sivilotti, "A Dynamically Configurable Architecture for Prototyping Analog Circuits", *MIT VLSI Conference*, pp. 237-258, 1988.

[9] E. Lee and G. Gulak, "A CMOS Field-programmable Analog Array", *ISSCC Digest of Technical Papers*, Feb., 1991, pp. 186-188

[10] K. Austin, "Integrated Circuit for Analog System", *U.S. Patent 5,196740*, Pilkington Micro-Electronics, March 23, 1993

[11] N. Sako, "Integrated Circuit and Gate Array", *U.S. Patent 5,298,806*, Kawasaki Steel Corp., March 29, 1994

[12] Bogdan Pankiewicz, Marek Wojcikowski et al., "A Field Programmable Analog Array for CMOS Continuous-Time OTA-C Filter Applications", *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 2, February 2002

[13] Precision Monolithics Inc., "Analog Signal Processing Subsystem", *GAP-01 Data Sheet* 1982

[14] F. Goodenough, "Analog Counterparts of FPGAs Ease System Design", *Electronic Design*, October 14, 1994

[15] http://www.zetex.com/3.0/a5-6.asp

[16] http://www.anadigm.com/

[17] Scree Ganesan and Ranga Vemuri, "A Methodology for Rapid Prototyping of Analog Systems", *12th Intl. Conf. VLSI Design*, pp.556-563, 1999

[18] Sree Ganesan and Ranga Vemuri, "FAAR: A Router for Field-Programmable Analog Arrays", *12th Intl. Conf. VLSI Design*, pp.556-563, 1999

[19] Behzad Razavi, "CMOS Technology Characterization for Analog and RF Design", *IEEE Journal of Solid-State Circuit*, Vol. 34, No.3, March 1999

[20] TSMC 0.18 *um* Process Design Kit, http://www.mosis.org

[21] J. Baker *et. al*, *CMOS circuit design, layout, and Simulation*, IEEE Press, 1997

[22] S. Trimberger, *Field-Programmable Gate Array Technology*, Kluwer Academic Publishers, 1994

[23] http://www.actel.com

[24] M. John and S. Smith, *Application-Specific Integrated Circuits*, VLSI Systems Series, 1997

[25] R. T. Smith, J. D. Chlipala, "Laser programmable Redundancy and Yield Improvement in a 64K DRAM," *IEEE J. Solid-State Circuits*, vol. SC-16, pp. 506-514, Oct. 1981.

[26] S. S Cohen and G. H. Chapman, "Laser Beam Processing and Wafer-Scale Integration," *Beam Processing Technologies*, Academic Press 1989.

[27] J. B. Bernstein, Y. Hua, and W. Zhang, "Laser energy limitation for buried metal cuts," *IEEE Elect. Dev. Let.*, vol. 19, no. 1, pp. 4-6, 1998.

[28] J. B. Bernstein, T. M. Ventura, and T. Radomski, "High Density Laser Linking of Metal Interconnect," *IEEE Trans. on Comp., Pack., and Manuf. Tech.*, Vol.17, pp. 590-593 Dec. 1994.

[29] J. B. Bernstein, B. D. Colella, "Laser-Formed Metallic Connections Employing a Lateral Link Structure," *IEEE Trans. on Comp., Pack. and Manuf. tech., Part A*, Vol.18, pp. 690-692, Sep. 1995.

[30] Y. L. Shen, S. Suresh, and J. B. Bernstein, "Laser Linking of Metal Interconnect: Analysis and Design Considerations," *IEEE Trans. on Elect. Dev.*, Vol. 43, pp. 402-410 Mar. 1996.

[31] J. B. Bernstein, W. Zhang and C. H. Nicholas, "Laser Formed Metallic Connections," *IEEE Trans. Comp. Pack. and Manuf. Tech., Part B: Advanced Packaging*, Vol. 21, No. 2, pp. 194, May 1998.

[32] J. Lee, *Analysis of Laser Processing of Metal Wires used in Microelectronics Applications*, Doctoral dissertation, University of Maryland, College Park, 2001.

[33] W. Zhang, J. Lee and J. Bernstein, "Energy Effect of Laser-induced Vertical Metallic Link", *IEEE Trans. Semiconduct. Manufact.*, 2001.

[34] W. Zhang, *Laser-induced Vertical metallic Link and Implementations in VLSI*, Doctoral dissertation, University of Maryland, College Park, 2000.

[35] http://www.enre.umd.edu/JB.

[36] K. Chung, J. Luo, H. Huang, J. Tuchman and J. Bernstein, "Experimental Verification of the Optimal Laser-Induced Advanced-Lateral MakeLink Structures", submitted to *IEEE Transaction of Semiconductor Manufacturing*, 2005.

[37] J. Luo, M. Peckerar, J. Bernstein, "A Novel Method to Reduece Op Amp/Comparator Offset", Patent Disclosure, University of Maryland 2004.

[38] Sree Ganesan and Ranga Vemuri, "FAAR: A Router for Field-Programmable Analog Arrays", *12th Intl. Conf. VLSI Design*, pp.556-563, 1999.

[39] Naveed A. Sherwani, *Algorithms for VLSI Physical Design Automation*, Kluwer Academic Publishers, 1999.

[40] V. Betz, J. Rose and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*, Kluwer Academic Publishers, 1999.

[41] T. Cormen, C. Leiserson *et. al*, *Introduction to Algorithms* McGraw-Hill, 2001.

[42] J. M. Ho, G. Vijayan, and C. K. Wong, "New algorithms for the rectilinear Steiner tree problem", *IEEE Tans. Computer-Aided Design*, vol. 9, no. 2, 1990.

[43] Y. Sun, T. Wang etc., "Routing for Symmetric FPGA's and FPIC's", *IEEE Tans. Computer-Aided Design of Integrated Circuits and Systems*, Vol. 16, No. 1, January 1997.

[44] G. Lemieux, S. Brown, D. Vranesic, "On Two-Step routing for FPGAs", *ACM Symp.on Physical Design*, 1997.

[45] S. Brown, J. Rose *et. al*etc., "A Detailed Router for Field-Programmable Gate Arrays", *IEEE Trans. on Computer-aided Design*, Vol. II, No. 5, May 1992.

[46] J. Rose, "Parallel Global Routing for Standard Cells", *IEEE Trans. On CAD'*, Oct. 1990.

[47] Y. Chang, S. Thakur *et. al*, "A New Global Routing Algorithm for FPGAs", *ICCAD*, 1994.

[48] C. Ebeling, L. McMurchie *et. al.*, "Placement and Routing Tools for the Triptych FPGA", *IEEE Trans. On VLSI*, Dec. 1995.

[49] G. Borriello, C. Ebeling, "The Triptych FPGA Architecture", *IEEE Trans. On VLSI Systems*, Vol. 3, No. 4, Dec., 1995.

[50] Y. Wu, M. Sadowska, "Routing for Array-Type FPGA's", *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, Vol. 16, No. 5, May 1997.

[51] L. McMurchie, C. Ebeling, "Pathfinder: A Negotiateion-based Performance-Driven Router for FPGAs", Univ. of Washington, 1996.

[52] C. Lee, "An Algorithm for Path Connections and its Applications", *IRE Trans. Electron. Comp*, Vol EC=10, 1961.

[53] J. Swartz, V. Betz and J. Rose, "A Fast Routability-Driven Router for FPGAs", *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, 1998.

[54] V. Betz, J. Rose, "VPR: A New Packing, Placement and Routing Tool for FPGA Research", *1997 International Workshop on Field Programmable Logic and Applications.*

[55] R. Kruse, C. Tondo, B. Leung, *Data Structures and Program Design in C*, Prentice Hall, 1997.

[56] U. Choudhury, and A. Sangiovanni-Vincentelli, "Constraint-Based channel routing for analog and mixed analog/digital circuits" *IEEE Tans. Computer-Aided Design of Integrated Circuits and Systems*, Vol. 12, No. 4, 1993.

[57] U. Choudhury, and A. Sangiovanni-Vincentelli, "Use of Performance Sensitivities in Routing of Analog Circuits", *CH2868-8/90, IEEE* 1995.

[58] W. Kao, Cy. Lo, M. Basel and R. Singh, "Parasitic Extraction: Current State of the Art and Future Trends", *Proceedings of the IEEE*, vol. 89, No. 5, May 2001.

[59] J.H. Chem, J. Huang, L. Arledge, P. C. Li and P. Yang, "Multilevel Metal Capacitances Models for CAD Design Synthesis Systems", *IEEE Elec. Dev. Letters*, 13 (1): 32-34, Feb. 1992.

[60] R. T. Edwards, K. Strohbehn and S. E. Jaskulek, "A Field-Programmable Mixed-Signal Array Architecture Using Antifuse Interconnects", *ISCAS* pp. 319-322, 2000.

[61] H. Alzaher and M. Ismail, "A CMOS Fully Balanced Four-Terminal Floating Nullor", *IEEE Trans. On circuit & Systems I: Fundamentals Theory & Applications*, Vol. 49, April 2002.

[62] J. Luo, H. Huang, J.B. Bernstein, J. Ari Tuchman and M. Peckerar, "A Configurable Analog Block Architecture for Field Programmable Analog Arrays", UMCP Patent Disclosure, 2004.

[63] A. Bratt and I. Macbeth, "Design and Implementation of a FPAA", *ACM/SIGDA FPGA'96*, Monterey, Ca., Feb. 11-13, pp. 88-93, 1996.

[64] H. Kutuk and S. Kang, "A Field-Programmable Analog Array (FPAA) Using Swithed-Capacitor Techniques", *IEEE ISCAS*, pp. 41-44, 1996.

[65] D. Vallancourt and Y. P. Tsividis, "Timing-Controlled Switched Analog Filters with Full Digital Programmability", *IEEE ISCAS*, pp. 329-333, 1987.

[66] Adaptive Logic, *AL220 Analog Micro Controller preliminary data sheet*, http://www.adaptivelogic.com

[67] S. T. Chang, B. R. Hayes-Gill and C. J. Paull, "Multi-Function Block for a Switched Current Field Programmable Analog Array", *Midwest Systemsium on Circuits and systems*, Ames, Iowa, August 18-21 1996.

[68] Sophocles J. Orfanidis, *Introduction to Signal Processing*, Prentice Hall,1996.

[69] E. Lee and P. G. Gulak, "Field-Programmable Analogue Array based on MOSFET Transconductors", *Electronic Letters 28(1)*, pp. 28-19, January 2, 1992.

[70] S. Chang, B. Hayes-Gill, and C. Paull, "Implemention of a Mulit-Function Signal Detection Block for a Field-Programmable Analogue Array", *Fifth Eurochip Workshop on VLSI Design Training*, Oct. 17-19, pp. 226-231, Dresden, Germany, 1994.

[71] E. Pierzchala, M.Perkowski, Paul Van Halen and Rolf Schaumann, "Current-Mode Amplifier-Integrator for a Field-Programmable Analog array", *ISSCC Disest of Technical Papers*, pp. 196-197, Feb. 1995.

[72] C. Premont, R. Grisel, N.Abouchi and J. Chante, "Current-Conveyor Based Field Programmable Analog Array", *Midwest Systemsium on Circuits and systems*, Ames, Iowa, August 18-21 1996.

[73] S. H. K. Embabi, X. Quan, N. Oki, A. Manjrekar and E. Sanchez-Sinencio, "A Field Programmable Analog Signal Processing Array", *Midwest Systemsium on Circuits and systems*, Ames, Iowa, August 18-21, 1996.

[74] J. Colinge, *Silicon-on-Insulator Technology: Materials to VLSI*, Kluwer Academic Publishers, 1991.

[75] P. Gray, P. Hurst, S. Lewis and R. Meyer, *Analysis and Design of Analog Integrated Circuits*, John Wilet & Sons, Inc., 2000.

[76] H. Alzaher and M. Ismail, "A CMOS Fully Balanced Four-Terminal Floating Nullor", *IEEE Trans. On circuit & Systems I: Fundamentals Theory & Applications*, Vol. 49, April 2002.

[77] K. Nakamura, "An 85 mW, 10 b, 40 Msample/s CMOS Parallel-Pipelined ADC", *IEEE J. of Solid-State Circuits*, Vol. 30, No. 3. pp. 629-633, March 1995.

[78] R. Whatley, "Fully Differential Operational Amplifier with DC Common-Mode Feedback", U.S. Patent 4, 573,020, Feb. 1986.

[79] C. Shih and P. Gray, "Reference Refreshing Cyclic Analog-to-Digital and Digitall-to-Analog Converters", *IEEE J. of Solid-State Circuits*, Vol. 21, No. 4, pp. 544-554, 1986

[80] M. Pelgrom, C. Duinmaijer and A. Welbers, "Matching Properties of MOS Transistors", *IEEE J. Solid-State Circuits*, Vol. 24, No. 5, Oct. 1989.

[81] J. Baker, *CMOS Circuit Design, Layout, and Simulation*, 2nd Edition Wiley-IEEE 2004

[82] Christian C. Enz, Gabor C. Temes, "Circuit Techniques for Reducing the effects of Op Amp Imperfections: Autozeroing, Correlated Double Sampling, and Chopper Stabilization", *Proceedings of the IEEE*, Vol. 84, No. 11, pp. 1584-1614, November 1996.

[83] B Razavi, Bruce Wooley, "Design Techniques for High-Speed, High-Resolution Comparators", *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 12, December 1992.

[84] http://www.analog.com

[85] D. Hilbiber, "A New Semiconductor Voltage Standard", *ISSCC Dig. of Tech. Paper*, pp. 32-33, February 1964

[86] *LM113 Data Sheet*, National Semiconductor Linear Data Book, 1972

[87] A.P. Brokaw, "A simple three-terminal IC bandgap reference", *IEEE Journal of Solid-State Circuits*, vol. SC-9, pp. 388-393, Dec. 1974

[88] J. Baker, *CMOS Circuit Design, Layout, and Simulation*, 2nd Edition Wiley-IEEE 2004

[89] M.A.T. Sanduleanu, A.J.M. van Tuijl, and R.F. Wassenaar, "Accurate low power bandgap voltage reference in 0.5 *um* CMOS technology," *IEE Electronics Letters*, vol. 34, pp. 1025-1026, 14th May 1998.

[90] V.G. Ceekala, L.D. Lewicki, J.B. Wieser, D. Varadarajan, and J. Mohan, "A method for reducing the effects of random mismatch in CMOS bandgap references," *Proc. IEEE Intl. Solid-State Circuits Conf.*, vol. 2, pp. 318-319, 2002.

[91] Y. Tsividis, "Accurate analyzes of temperature effects in $I_C$-$V_{BE}$ characteristics with application to bandgap reference sources", *IEEE J. Solid State Circuits*, vol. 15, pp. 1076-1084, Dec. 1980.

[92] B. Song and P. Gray, "A Precision Curvature-Compensated CMOS Bandgap Reference", *IEEE J. Solid State Circuits*, vol.sc-18, No.6, 1983.

[93] I. Lee, G. Kim and W. Kim, "Exponential Curvature-Compensated BiCMOS Bandgap References", *IEEE J. Solid State Circuits*, Vol. 29, No. 11, Nov., 1994

[94] P. Malcovati, F. Maloberti *et. al*, "Curvature-Compensated BiCMOS Bandgap with 1-V Supply Voltage", *IEEE J. Solid State Circuits*, Vol. 35, No. 7, July 2001.

[95] M. Gunawan, G. Meijer, J. Fonderie, and H. Huijsing, "A curvature corrected low-voltage bandgap reference", *IEEE J. Solid-State Circuits*, vol. 28, pp. 667-670, June 1993.

[96] R. Schaumann, Mac E. Valkenburg, *Design of Analog Filters*, Oxford University Press, 2001.

[97] R. Baker, *CMOS Mixed-signal Circuit Design*, IEEE Press, 2003.

[98] , "A Practical Method of Designing RC Active Filters", *IRE Trans. Circuits Theory*, Vol. CT-2, No. 3, pp. 74-85, 1955.

[99] H. Alzaher and M. Ismail, "A CMOS Fully Balanced Four-Terminal Floating Nullor", *IEEE Trans. on Circuit and System II: FUNDAMENTAL THEORY AND APPLICATIONS*, VOL. 49, NO. 4, APRIL 2002.

[100] R. Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, 2nd Edition, Kluwer Academic Publishers, 2003.

[101] B. Razavi, *Principles of Data Conversion System Design*, IEEE Press, 1995.

[102] M Choi and A. Abidi, "A 6-b 1.3-Gsample/s A/D Converter in 0.35*um* CMOS", *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, VOL. 36, NO. 12, DECEMBER 2001.

[103] P. Scholtens and M. Vertregt, "A 6-b 1.6-Gsample/s Flash ADC in 0.18*um* CMOS Using Averaging Termination", *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, VOL. 37, NO. 12, DECEMBER 2002.

[104] M. Choe, B. Song, K. Bacrania, "A 13-b 40-MSample/s CMOS Pipelined Folding ADC with Background Offset Trimming", *IEEE J. Solid-State Circuits*, vol. 35, pp. 1781C1790, Dec. 2000.

[105] P. Vorenkamp, R. Roovers, "A 12-b, 60-MSample/s Cascaded Folding and Interpolating ADC", *IEEE J. Solid-State Circuits*, vol. 32, pp. 1876-1886, Dec. 1997.

[106] J. Shieh, M. Patil and B. Sheu, "Measurement and Analysis of Charge Injection in MOS Analog Switches", *IEEE J. Solid-State Circuits*, Vol. 22, No. 2, April 1987.

[107] G. Wegmann, E. Vittoz and F. Rahali, "Charge Injection in Analog MOS Switches", *IEEE J. Solid-State Circuits*, Vol. 22, No. 6, December 1987.

[108] P. Li, M. Chin, P. Gray and R. Castello, "A Ratio-Independent Algorithmic Analog-to-Digital Conversion Technique", *IEEE J. Solid-State Circuits*, Vol. SC-19, No. 6, December 1984.

[109] Design of the 4-bit DAC and Encoder from H. Huang and K. Laurentz, ASDL Lab, Department of Electrical and Computer Engineering, University of Maryland, College Park